AD A144829

0

COMPUTER AIDED CONCEPTUAL DESIGN

OF SUBMARINES

by

PATRICK CHARLES HALE

B.S., University of Washington
(1975)

M.B.A., National University
(1981)

N66314-70-A-0073

Submitted to the Department of
Ocean Engineering
in Partial Fulfillment of the
Requirements of the Degrees of

OCEAN ENGINEER

and

MASTER OF SCIENCE IN

NAVAL ARCHITECTURE AND MARINE ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1984

Signature of Author: Patrick C. Hale
Department of Ocean Engineering
11 May 1984

Certified by: David V Burke
Thesis Supervisor

Accepted by: A. Douglas Carmichael
Chairman, Ocean Engineering
Departmental Committee

COMPUTER AIDED CONCEPTUAL DESIGN OF SUBMARINES

by

PATRICK CHARLES HALE

Submitted to the Department of Ocean Engineering on May 11, 1984 in Partial Fulfillment of the Requirements of the Degrees of Ocean Engineer and Master of Science in Naval Architecture and Marine Engineering

ABSTRACT

A software package was developed to perform conceptual design of submarines, using the Computervision CGP-200X Designer System, a turn-key computer aided design hardware and graphics software system. The philosophy behind the software package is to keep all major design decisions under the control of the design engineer, rather than embedding key decisions in the program algorithms.

Modules are provided for calculating weight estimates, principal characteristics and envelope geometry, resistance, weight and moment balance, and the equilibrium polygon. The package interfaces with a pressure hull design module developed separately in an O.E. thesis by Marvin Meade. Interactive graphics are used where appropriate.

The software requires a knowledgeable naval architect as the user, but does not require extensive knowledge of computers or computer aided design systems.

Thesis Supervisor: Dr. David V. Burke

Title: Professor of Ocean Engineering

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This thesis is the product of many people's work and support in addition to my own. Mr. Richard Salter, of Computervision Corporation, was instrumental in providing unlimited system time and initial training on the Computervision Designer System. Mr. James Kelly, also of Computervision, freely gave of his time and exceptional knowledge of the system, saving the author countless hours in programming and testing the software package. Captain Harry Jackson, U.S. Navy, retired, was the inspiration for the concept of the entire thesis project, through his seemingly unlimited knowledge of submarine design, and his infectious enthusiasm for the entire subject area. Finally, Captain (Professor) David Burke, U.S. Navy, in his capacity as Thesis Supervisor, provided valuable guidance and insight, through his considerable submarine design experience, and Navy and academic professionalism.

The author gratefully acknowledges the contributions of these individuals, without which the successful completion of this thesis would not have been possible.

INTRODUCTION

GENERAL DISCUSSION

Computer Aided Design (CAD) has become an important tool in many industries over the past ten years. The capabilities of CAD systems are growing at a rate which defies efforts to catalog them. In the aerospace and automotive industries, as well as in large architectural firms, CAD has become the standard mode of design.

In the area of naval ship design, CAD has been adopted more slowly, for a number of reasons. The comparatively low production rates bring the issue of cost effectiveness into greater question for ship design agencies. The tremendous complexity and scale of naval combatants makes the design of a flexible and comprehensive package of software a formidable task.

Despite these problems, a number of packages for conceptual and preliminary design of surface combatants have been implemented in both government and private sector design organizations. The packages used by

government agencies have tended to be batch-oriented ship synthesis programs with little or no graphic output, and virtually no real-time interactive graphic design capability (the notable exception to this is the use of interactive interior ship layout software).

This lack of interactive graphics has been partially hardware driven; the concern for standardization and the lengthy process of government certification and procurement cause a substantial delay between the initial availability and subsequent installation of new hardware. In addition, the CAD process is foreign to many of the prominent designers in the very tradition-oriented world of naval architecture. Many design procedures in naval architecture are somewhat subjective and difficult to quantify.

A further impediment to the use of CAD has been the tendency of many early software packages to usurp designer perogative by having key decisions in the design process "hard-wired" into the program algorithms. Such software does not utilize the expertise of the designer, and frequently precludes trading off design parameters to achieve optimal designs.

In the area of submarine design, the above problems are further accentuated. The volume of new submarine designs is even lower than that of surface combatants. Accuracy becomes critical; where feet and tons may be acceptable units for tolerance in large surface ship

designs, inches and pounds are frequently the desired accuracy level in submarine design. This is due to the hydrostatic requirement to balance both surfaced and submerged, as well as to the premium attached to internal volume and deck surface area.

Submarine designers are fewer in number than their surface counterparts, and they often have many design relationships tied to subjective criteria (i.e., "designer's eye") developed over years of experience. A program which does not allow the submarine designer to retain this subjective design flexibility is not likely to be accepted by the design community.

SUBMARINE CAD: A HISTORICAL PERSPECTIVE

Although Computer Aided Design has not been a primary tool in submarine design, numerous programs have been developed to accomplish analytical chores that are subsets of the overall design process. These include propellor design, resistance estimates, hydrostatic calculations, and structural design. These programs have been primarily analytical in nature, providing no interactive graphic capability, and are usually written for main-frame computer systems.

Several packages have been written to achieve a complete iteration of the design process. These include the CODESUB program [1]. It provides numerical, rather

than graphic, output. CODESUB was developed by the Center for Naval Analyses in order to aid in projecting future submarine design characteristics for both our navy and its potential adversaries. This program was not intended to be a detailed tool for the naval architect, and, unfortunately, has many key design parameters imbedded in the source code, which restricts its use in trading off alternative designs. Other packages "graft" together sections of existing designs, matching the largest hull diameter, to provide a composite design. ASSET [2], developed by Boeing Computer Services, while primarily a surface ship design package, may be expanded to include a submarine design module in the future. ASSET provides some low-level graphics output, but is not a full, real-time interactive graphics package.

Thus, there is a clear opportunity for the development of a submarine design package incorporating the use of the full capabilities of currently available CAD systems. This thesis represents an exploration of such a design package.

PHILOSOPHY FOR THE THESIS PROGRAM

The software developed in this thesis accomplishes a "first pass" through the submarine design process at the conceptual design level. In conjunction with the concurrent Ocean Engineer thesis written by Marvin

Meade, the package includes calculation of weight estimates, envelope and pressure hull geometry, resistance (speed and power), basic hydrostatics, weight and moment balance, and determination of the equilibrium polygon.

The primary philosophy of this package is to develop it for a user who is a competent naval architect, but who may not necessarily be an expert in the use of computers. Crucial design decisions are left to the user, rather than being embedded in the program code. Opportunities are provided to override program algorithms where necessary to implement the desires of the individual designer. In addition, if the designer does posess some knowledge of CAD systems or computer programs, it is easy to exit the package at appropriate points and tailor the design to provide more detail or alternate geometry and analytical procedures. Interactive graphics are used to provide a clear visualization of the design under development. All program input parameters are couched in standard naval architecture terms to facilitate the user's interaction with the package. The title chosen for the software package is CADSUB (Computer Aided Design of Submarines).

Input to each program module is explained in the chapter documenting the particular module. Samples of program output are provided in Appendix I to this thesis.

HARDWARE:

THE COMPUTERVISION DESIGNER SYSTEM

GENERAL DISCUSSION

The optimum choice of hardware for implementing the philosophy of this thesis was one of the several high-quality "turnkey" CAD systems, all of which permit the use of interactive graphic design techniques. Graphics packages for main-frame systems are available from several prominent suppliers, but are a compromise at best, since the processors they run on are not optimized for graphics-intensive applications.

Suitable systems are marketed by several suppliers, including Computervision, Applicon, Bendix, McDonnell-Douglas Automation Division, and others. For this thesis, Computervision Corporation, of Bedford, Massachusetts, offered free system time, instruction, and other technical aid to develop the program.

THE COMPUTERVISION DESIGNER SYSTEM

The system which Computervision provided time and instruction on is their basic "Designer System", built around the CGP-200X processor. The heart of the system consists of the processor, a high-speed tape drive, and one or two 300 megabyte hard disk units. The CPU is available with several increments of random access memory (RAM). The particular system used for this thesis was configured for approximately 1.3 megabytes of core memory. The architecture of the CGP-200X is 16-bit.

The Designer System utilizes two distinct operating systems. Basic file management and system-level housekeeping chores are handled at the "OS" level, which is similar in use to the operating systems on typical mini-computer installations. For the graphics operating system, Computervision uses the "CADDS 4-X" environment, which is tailored for graphics, and automatically interfaces with a powerful data base management system. Graphics commands are entered in a simple verb-noun syntax, such as "INSERT LINE" or "INSERT SPLINE".

Input/output (I/O) is handled by several devices. For system administration and text file input, small alpha-numeric terminals are provided. For graphics I/O, high-resolution color or monochrome terminals are provided as part of a comprehensive workstation, which

includes a thermal printer for working quality text and graphics printout. Also provided at each workstation are a digitizing tablet and a display control device. This device controls background and graphics intensities, the number of lines of text displayed (4 or 24 lines), and dynamic control of the display; zooming, rotation, and scrolling. Output of finished quality is available from line printers, color pen plotters, and black and white electrostatic plotters. Large format output is supported by the plotters.

PROGRAMMING LANGUAGES

Several languages are provided for use with the Designer System. The primary language is Fortran-S, a subset of ANSI Fortran. System subroutines are provided for interacting with the graphics and data base management operating systems. Programs entered in Fortran-S execute with the greatest speed, and allow creation of new graphics commands. Because of limited CPU space, the utilities required for this level of programming, which are often provided on main-frame systems, must be separately programmed for each module. Such utilities include linking and loading capabilities. Because of the length of training required to do competent programming at this level, this thesis was programmed in an alternate language.

Two macro languages are provided for the system. Both allow creation of execute files using standard graphics commands. VARPRO2 is a non-compilable macro language which supports Fortran-like I/O and computational statements. It is a flexible and effective language, but executes rather slowly. NEWVAR, the other macro language provided, is compilable, and therefore executes far more rapidly. The I/O for NEWVAR is cumbersome and difficult to format. All I/O must be handled in text string form, and converted within the program to numerical values. Despite this shortcoming, the execution speed of NEWVAR led to its adoption for use in this thesis. Ultimately, the algorithms in this thesis should be translated to Fortran-S for maximum speed and I/O flexibility.

PRELIMINARY WEIGHT ESTIMATES

IMPORTANCE OF WEIGHT ESTIMATES

The algorithms employed in this thesis assume a weight-driven submarine design. This assumption means that the envelope size calculated for the submarine will contain enough volume to displace the same amount of water, when submerged, as the weight of the envelope displacement for the submarine. If the desired design is known or suspected to be volume-limited, an arbitrary excess of lead may be specified during the weight calculations, thus driving the program to calculate a larger envelope size. The designer may translate this additional weight into volume manually to allow for a volume-limited design.

Since the estimated weight data will drive the basic dimensions of the submarine, the weight estimating algorithms must be made as accurate as possible. The designer must carefully consider the input data to the weight module, and, where appropriate, may decide to

override the programmed algorithms and specify the value for a particular weight group. The program is designed to automate the calculations, and not the design judgement, for the user. A knowledgeable user may easily modify the algorithms themselves to suit particular needs.

WEIGHT ESTIMATING ALGORITHMS

The accounting system chosen for weight estimation is the SWBS (Ship Work Breakdown Structure) convention used by the Naval Sea Systems Command for ship weight records [3]. This system consists of seven major numbered groups for all ships. For submarine weight reports, several special weight categories are added. A brief summary of the weight groups is as follows:

GROUP	DESCRIPTION
I	Structural
II	Propulsion
III	Electrical
IV	Combat Systems
V	Auxiliary Systems
VI	Outfit and Furnishings
VII	Armament
A1	Sum of I-VII
Lead	Total lead
A	A1 + Lead
Variable Load	(fuel, stores, etc.)

NSC	Normal Surface Condition
Reserve Buoyancy	Main Ballast
Submerged Disp.	Self explanatory
Free Flood	Non-buoyant flooded volume
Envelope Disp.	Submerged Disp. + Free flood

The algorithms chosen for the weight estimating module take standard design parameters as input, where possible, and provide an output in terms of a percentage of A1 weights, NSC, or submerged displacement, as appropriate. These percentages are then combined with the discrete weights that are calculated or input for selected weight groups to determine the A1 weight total, lead, NSC, and other weight parameters. The designer has the option of specifying weights for groups II, IV, VI, and VII, which override the programmed algorithms. The following section discusses the chosen algorithms for calculating each weight group. Throughout this thesis, the following conventions will be used to indicate mathmatical operations:

"+" = addition

"-" = subtraction

"*" = multiplication

"/" = division

"^" = exponentiation

GROUP I:

Weight group I is determined as a percentage of

NSC. In general,

$\%W1 = f(\text{depth, material, size})$.

The size dependence is accomodated by calculating W1 as a percentage of NSC. The algorithm implemented in the program is:

$\%W1 = C1 + C2 * DEPTH$.

C1 and C2 are dependent on hull material used, and are determined parametrically from data on past submarine designs [4,5]. The DEPTH parameter is a user input of maximum operating depth in feet. The accepted design practice of designing for a collapse depth of 150% of operating depth is implemented through the choice of C1 and C2.

GROUP II:

Weight group II is calculated directly, in tons, rather than as a percentage. The number of parameters involved in calculating W2 is greater than for any other weight group.

W2=f[horsepower, propulsion type, battery type & capacity (capacity=endurance)]

The horsepower variable for a nuclear plant is simply shaft horsepower, while, for a non-nuclear plant, two separate horsepowers are needed; shaft horsepower (usually the horsepower of the electric final drive motor) and charging horsepower (power of the prime mover used to drive the alternators or generators to provide

electrical power to recharge the storage batteries). The weight of the charging prime mover is a strong function of technology chosen; ie diesel, wankel, etc.

The calculation of storage battery weight for a non-nuclear plant is dependent on battery type, capacity in kilowatt-hours at a high discharge rate ("sprint"), and capacity in KW-H at a low discharge rate (endurance power). The options provided for battery type in this program are nickel-cadmium, improved lead-acid (German VARTA type), and "standard" lead acid (U.S. Trident type). Constants determined from the energy densities of these battery types at the two discharge rates are used to determine battery weight (the larger of sprint or endurance rate battery weights is chosen as the final battery weight).

Electric motor weight is determined from a multiplier based on a modern, air-cooled DC motor. The option exists to provide other multipliers based on liquid cooling, superconducting technology, etc.

For the non-nuclear plant, an additional option exists to provide a lumped adjustment to weight group II to provide for unconventional propulsion technology (fuel cells, Stirling, low-power nuclear, etc.). This final adjustment allows the designer maximum flexibility to tailor group II without making modifications to the program code.

For a "straight" nuclear plant,

W2=(C1*SHP)/[log(SHP)]^C2

where SHP=shaft horsepower

C1 is from parametric data.

For a non-nuclear plant,

W2=WE+WM+WB+dW2

where WE=charging prime mover weight

WM=propulsion motor weight

WB=battery weight

dW2=adjustment to W2 (user input)

WE=C3*CHP

where CHP=charging horsepower

C3 is dependent on prime mover type.

WM=C4*SHP.

WB=Max[(C5*KWHS),(C6*KWHE)]

where KWHS=capacity for "sprint"

KWHE=capacity for endurance

C5,C6 are dependent on battery type.

GROUP III:

Although weight group III would intuitively appear to be a direct function of installed generating capacity, a study of historical design data [5] reveals that the group III weights are, in fact, very closely approximated by a straight percentage of A1 weights. This anomaly is best explained by the large amount of this weight group attributable to the electrical distribution system spread throughout the submarine.

This power distribution system weight is a direct function of the size of the submarine, since the amounts of cable, connectors, and distribution panels vary with size of the vessel. Weight group III, consequently, is calculated as a percentage of A1 weights:

%W3=C1.

GROUP IV:

Determination of group IV weights is very difficult to implement in an "automatic" mode. The variety of combat system components and great variation in equipment weights makes the selection of an algorithm extremely difficult, as most schemes provide accurate results for a limited number of combat suites, at best. It is therefore highly recommended that the designer specify a group IV weight based on his off-line determination of the desired combat systems and associated weights. To permit a rough approximation is acceptable for an early design iteration, the program contains an algorithm based on historical data as a percentage of A1 weights for either fast attack or ballistic missile submarines:

%W4=Ci i=1,4

where Ci is based on submarine type (SS, SSB, SSN, or SSBN).

GROUP V:

Weight group V is accurately determined as a function of submarine size, since the auxiliary systems required for a vessel are directly proportional to displacement and internal volume. Group V is thus calculated as a percentage of A1 weights:

$\%W5=C_i \quad i=1,2$

where Ci is based on submarine type (attack or ballistic missile).

GROUP VI:

Group VI weights are a function of crew size, and, to some extent, typical mission duration. The multiplier for group VI weight determination is based on habitability standards commensurate with the last two classes of attack and ballistic missile submarines, respectively. The weights are directly calculated in tons:

$W6=C1*NP$

where NP=number of personnel in the crew.

GROUP VII:

The estimation of armament weights is nearly as difficult as the task of determining group IV weights, since the possible combinations of existing and future weapons and associated launchers are infinite. Again, it is highly recommended that the designer specify a value

for weight group VII based on his off-line determination of the weapons payload. As was provided in the case of combat systems, a rough estimate of group VII is available from a percentage of A1 weights. The percentage constant was determined from historical data, and separate values are provided for attack and ballistic missile submarines.

%W7=C1.

OTHER WEIGHTS:

The amounts of lead, variable load, reserve buoyancy, and free flood are provided as fractions by the designer. The following ranges are appropriate:

Lead	(.07-.11)*A1
Variable Load	(.04-.07)*NSC (nuclear)
	(.08-.19)*NSC (non-nuclear)
Reserve Buoyancy	(.12-.15)*NSC
Free Flood	(.05-.1)*(Submerged Displ.)

The operator, however, may specify his own values as desired.

CALCULATION OF WEIGHT SUMMARY

When either percentages or actual values have been calculated or specified for each of the weight groups, the calculations for the weight summary are processed. At a minimum, actual weight values in tons have been

determined for weight groups II and VI. Additionally, the designer may have specified weights for weight groups IV and VII. As an example of the algorithms used for final weight calculations, assume that only W2 and W6 have been determined in tons. The other permutations of final weight calculations are similar, but allow for dicrete weights, vice percentages, for either W4, W7, or both. For the case where only W2 and W6 are known, the sequence is as follows:

(All "%" figures used are converted to fractions; %X=%X/100)

K1=(%W1+%W1*%LD)/(1-%VL)+%W3+%W5

where %LD=lead fraction

%VL=variable load fraction.

A1=(W2+W6)/(1-K1-%W4-%W7)

W3=%W3*A1

W4=%W4*A1

W5=%w5*A1

W7=%W7*A1

Lead=%LD*A1

A=A1+Lead

NSC=A/(1-%VL)

Variable Load=%VL*NSC

Main Ballast=%RBY*NSC

where %RBY=fraction of reserve buoyancy

Submerged Disp.=NSC+Main Ballast

Free Flood=%FF*Submerged Disp.

where %FF=free flood fraction

Envelope Disp.=Submerged Disp.+Free Flood.

The calculated weight summary is displayed on the screen with a summary of user inputs, at which time the designer may accept or reject the results. If the results are rejected, the program loops back to the input section for the weight module and another iteration is started. If the results are accepted, they are written into an output file for record purposes, and various input and calculated parameters are written into several intermediate files for use in later program modules as the design is completed.

Because of the high degree of importance attached to the calculated weight data, future refinement of program algorithms should include a priority effort to accomplish two major tasks:

(1) Better accuracy of calculated weights.

(2) Further subdivision of weights (at least partially to the three-digit SWBS level).

PRINCIPAL CHARACTERISTICS AND ENVELOPE GEOMETRY

PRINCIPAL CHARACTERISTICS

The results from the weight module provide the first important input to the determination of the remainder of principal characteristics, which are geometric in nature. Since this thesis implements a weight-driven submarine design process, the next design task is to create an envelope of the appropriate volume to contain the envelope displacement:

Envelope Volume=35*Envelope Displacement.

The product envelope defined by this module of the program will exhibit the following geometric principal characteristics (abbreviations and symbols in parentheses are program variable names):

Length Overall (LOA)

Length of Entrance (Lf)

Length of Run (La)

Mid Body Length (Lm)

Diameter (D)

Length/Diameter Ratio (L/D)

Prismatic Coefficient (Cp)

Bare Hull Surface Area (SF).

In addition, once envelope geometry is defined, the following calculated hydrostatic characteristics may be found:

Submerged Longitudinal Center of Buoyancy (LCBSUB)

Draft @ Normal Surface Displacement (TNSC)

Longitudinal Center of Buiyancy @ NSC (LCBNSC).

These parameters, then, when coupled with the input parameters delineated for the weight estimation module, constitute a set of principal characteristics for the design.

DEFINING THE ENVELOPE

A number of algorithms have been suggested to calculate the geometry of the envelope for submarines. These range from a "cut-and-paste" approach using past designs [2] to very sophisticated polynomial determination of offsets [6]. In between these extremes are a number of methods utilizing simple thumbrules and parametric relationships among Cp, D, L/D, and displacement [4,7].

Both the cut-and-paste and thumbrule approaches are somewhat imprecise in allowing the designer to fully

control the geometry of the resulting hull form. The polynomial methods, while very precisely controlled, require the designer to specify design parameters which are not couched in, or clearly related to, traditional naval architecture vocabulary. The most desirable algorithm, then, will allow strict control of the resulting geometry while accepting input parameters which are common naval architectural terms.

CHOSEN ENVELOPE GEOMETRY ALGORITHM

In choosing the algorithm implemented in this thesis, two restrictions were accepted for the sake of simplicity. The envelope shape is confined to circular cross-sections, and an analytically smooth envelope outline is calculated. Figure 4.1 demonstrates the nomenclature and subdivision of the envelope calculated with the chosen algorithm.

The designer is prompted for the following geometric input parameters:

Length of Entrance

Length of Run

Forebody Prismatic Coefficient (CpF)

Aft Body Prismatic Coefficient (CpA)

Maximum Diameter (D).

The algorithm then defines an envelope consisting of a hemi-elliptical forebody, a cylindrical midbody,

and a parabolic afterbody. The chosen algorithm is a modification of one suggested by Jackson [4].

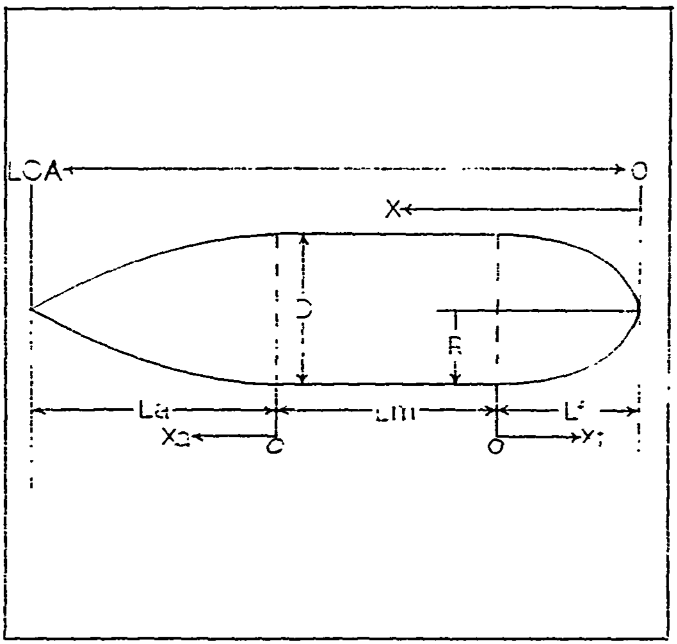


Figure 4.1: Nomenclature for determination of envelope geometry.

The coefficients Nf and Na are essentially fullness coefficients for the fore and aft bodies, respectively. In order to facilitate input in traditional naval architectural terms, these fullness coefficients are calculated from the values specified by the designer for prismatic coefficients of the fore and aft body sections of the envelope:

Nf= 71.0477*CpF^5 +65.8107*CpF^4 -483.3037*CpF^3 +587.4137*CpF^2 -281.7224*CpF +49.5876

Where Nf is the forward fullness coefficient.

Na= 379.6546*CpA^5 -938.4708*CpA^4 +944.8853*CpA^3 -471.0872*CpA^2 +119.1465*CpA -11.3454

Where Na is the aft fullness coefficient.

These polynomial relationships are valid for values of CpF from .5383-.8720, and for CpA from .3333-.7111. These ranges substantially exceed those for actual submarine designs. The relationships were derived by using known (hand-verified) equivalencies of prismatic and fullness coefficients to set up matrix equations with the polynomial coefficients as the unknown column vector. Gaussian elimination was then used to solve for the unkown coefficients. Additional hand-calculated equivalencies were then utilized to verify that, within the specified ranges, the polynomial relationships yield results correct to the fourth decimal place.

The "X" coordinate for the overall envelope runs from 0 at the forward perpendicular to LOA at the aft

perpendicular. In the formula for the forebody hemi-ellipse, a local X coordinate (Xf) runs from 0 at the end of the entrance length (Lf) to Lf at the forward perpendicular:

$$Xf = Lf - X$$

In the aft body parabolic section, a local X coordinate is again used:

$$Xa = X - (Lf + Lm)$$

Note that Lm, the mid body length, is a calculated parameter, and not an input. The calculation of Lm will be described later in this chapter.

The hull radius, r(X), for the elliptical forebody is calculated as follows:

$$r(X) = R*[(1-(Xf/Lf)\text{^}Nf)\text{^}(1/Nf)]$$

where R= D/2

The hull radius within the mid body is simply:

$$r(X) = R$$

The hull radius within the aft body parabolic section is:

$$r(X) = R*[1-(Xa/La)\text{^}Na]$$

The offset r(X) is calculated for each of 21 stations evenly spaced along the hull length. Obviously, r(X) at both the forward and aft perpendiculars is zero. When the offsets for the 21 stations have been determined, the sectional area of each station is calculated:

$$SA(X) = Pi*r(X)\text{^}2$$

The station sectional areas are then integrated along the length to determine the volume enclosed by the envelope:

$$VOL= \int_{0}^{LOA} SA(X)\, dX$$

This and all other integrations in this thesis are done numerically, using Simpson's Rule.

On the first pass through this process, the mid body length is zero, and LOA= Lf+La. The resulting volume, then, is the total volume for the fore and aft body sections alone. To determine the proper mid body length, the volume determined above is subtracted from the required envelope volume, and Lm is calculated as the length of a cylinder of radius R containing the remaining volume:

DV= Envelope Volume-VOL

Lm= DV/(Pi*R^2)

If DV is negative, an error message is printed, and the designer is prompted for new geometric input values.

After Lm is calculated, the final value of LOA is determined:

LOA= Lf+Lm+La

The sectional areas for the resulting hull form are calculated at the new 21 stations, after which circumferences and bare hull surface area are determined:

Circ(X)= 2*Pi*r(x)

$$SF= \int_0^{LOA} Circ(X)\ dX$$

The overall prismatic coefficient is calculated by comparing the envelope volume with the volume of a cylinder (Vcyl) of length LOA:

Vcyl= Pi*R^2*LOA

Cp= Envelope Volume/Vcyl

These calculated parameters, when taken together with the input values, constitute the geometric definition of the envelope.

CALCULATION OF BASIC HYDROSTATICS

In order to later balance the submarine, it is important to accurately determine the longitudinal centers of buoyancy in the submerged (LCBSUB) and normal surface (LCBNSC) conditions. Calculation of LCBNSC requires the determination of the draft at NSC. LCB's in both conditions are calculated by integrating the first longitudinal moment of the sectional areas.

Using the sectional areas for the fully-submerged hull as calculated above, LCBSUB is determined:

$$LCBSUB= \int_0^{LOA} X*SA(X)\ dX\ /VOLENV$$

There are several possible approaches to determining the draft at NSC. When the calculations are done manually, it is customary to choose at least three

points, commonly at drafts equal to D, D/2, and D+R/2, and determine the volumes and resulting displacements for each of these drafts. A curve of displacement versus draft is then constructed, after which the draft corresponding to the NSC displacement is read from the curve. While this method is expeditious and acceptable for hand calculations, relatively large errors may result when the designer is not experienced enough to draw the proper curve through the three points. Use of additional points improves the accuracy, but the manual calculations are tedious and time-consuming.

With the speed and accuracy of the computer available, a much more precise determination of draft at NSC (TNSC) is possible. A potentially important factor that is often overlooked in manual methods is the volume of the free flood, which is contained in the envelope, and may be a signifigant portion (2-10%) of the envelope volume. Study of existing designs indicates that approximately 80% of the total free flood volume is submerged at TNSC. Consequently, the "target" displacement in determining TNSC is not simply NSC, but rather includes this portion of the free flood:

NSCVOL= 35*(NSC+.8*Free Flood)

(for 80% free flood). Where NSCVOL is the desired volume at TNSC.

The method chosen to determine NSC draft is a bracket and halve iteration on the variable TNSC,

beginning at TNSC= 3*R/2. The allowable error in NSC volume is specified as 0.5% in the current program. This tolerance may be decreased, with additional iterations required for convergence. The value of 0.5% was chosen as sufficient for the conceptual design phase, while minimizing the number of iterations required. The algorithm employed is as follows:

DV= .005*NSCVOL

NPLUS= NSCVOL+DV

NMINUS= NSCVOL-DV

DR= .5*R

TNSC= R+DR

For each trial TNSC, the value of each station radius, r(X), is compared to DR:

If (DR)>r(X), then r(X) and SA(X) remain unchanged.

If (DR)<r(X), then:

SA(X)= SA(X) -[r(X)^2 *ARCOS(DR/r(X)) -DR *(r(X)^2 -DR^2)^.5]

(Zero trim is assumed).

$$VOL= \int_0^{LOA} SA(X)\, dX$$

If VOL>NMINUS and VOL<NPLUS, then the correct TNSC is determined. If VOL<NMINUS, then DR is increased by .5*DR. Similarly, if VOL>NPLUS, then DR is decreased by the same increment. In either case, the algorithm is repeated until the value of VOL is acceptable. When the

correct TNSC is found, LCBNSC ic calculated:

$$LCBNSC= \int_0^{LOA} X*SA(X)\ dX\ /NSCVOL$$

The calculated results are displayed on the screen for the designer's approval. If the results are deemed unsatisfactory, the designer may reject the envelope geometry and return to the input section for another envelope design iteration.

If the results are accepted, all of the specified input values and calculated data are written to an output file for record purposes, and appropriate data for other program modules are written to passing files for later use.

PLOTTING THE ENVELOPE

An accurate plot of the envelope is imperative, since the outline displayed on the monitor will guide the designer in his interactive design of the pressure hull. The 21 stations determined thus far are insufficient to fit an accurate curve, since the arbitrary equally-spaced stations may not fall at the points of greatest hull curvature, or at the junctions of the mid body with the fore and aft body sections.

To adequately portray hull curvature and transitions at section junctions, the program calculates 132 unequally-spaced offsets to be used in plotting the

envelope. Offsets are most closely spaced near the forward perpendicular and at the transition points of the hull. These points are then used to fit a Bezier spline curve which forms the profile outline of the envelope. The plot is automatically scaled to make full use of available screen area, and both vertical and horizontal scales are drawn as a reference for the designer.

The two-dimensional envelope outline is rotated to form a surface of revolution, and a mesh pattern is superimposed on the surface to aid visualization of the resulting three-dimensional hull form. In addition, an isometric view is displayed below the profile for perspective.

The resulting display of two views of the outer hull constitutes the graphic output of the envelope geometry module. The designer is now ready to proceed to the speed and power calculations.

SPEED AND POWER

GENERAL DISCUSSION

The calculation of resistance for the envelope geometry developed in the previous module is a straightforward analytic process. The formulae employed are the familiar drag estimation relationships from basic hydrodynamics. A proposal for future implementation would be the addition of a propellor design module to accurately predict the propulsive coefficient, in conjunction with the envelope geometry.

To facilitate endurance calculations for non-nuclear designs, the calculation of total electrical load at each speed is included. This total load figure must include all non-propulsion loads (hotel and combat system) as well as those required to propel the submarine.

With the data passed from the envelope geometry module, an estimate of appendage drag may be made. This, when combined with hull form and surface area data,

completes the input to the speed and power algorithm.

ESTIMATION OF APPENDAGE DRAG

Study of existing submarines reveals that appendage drag comprises thirty percent or more of the total drag on the submarine for most designs. In the later stages of the design process, detailed consideration should be given to the configuration and size of each appendage to improve the accuracy of resistance estimates. At the stage of conceptual design, however, details concerning appendages are often uncertain. Many designs are formulated without final selection of control surfaces, for example. The required shape, location and sizes of surfaces for cruciform, X, and inverted Y sterns may exhibit some variance, and the final configuration may not be chosen until much later in the design process.

In view of this uncertainty, a reliable rough estimate of total appendage drag is desirable for the conceptual design. This thesis incorporates the method suggested by Bukalov [8]. Bukalov studied numerous actual submarine designs, and developed an algorithm based on this parametric study. This algorithm may be reduced to a single formula dependent on length overall (LOA) and maximum diameter (D). The appendage drag (Da) is then calculated as follows:

Da= 1.09065E-3*LOA*D+11.25

CHOSEN ALGORITHM

The basic relationships implemented in this program module may be found in numerous introductory texts [9,10]. The interactive input from the terminal consists of the following items:

Propulsive Coefficient (PC)

Non-propulsion Loads in KW (HL)

The remainder of the required input is passed from the envelope geometry module, and consists of:

Prismatic Coefficient (Cp)

Length Overall (LOA)

Maximum Diameter (D)

Bare Hull Surface Area (SF)

The speed and power data is calculated and stored in an array indexed on speed in knots (Vk) from zero to forty knots. After appendage drag is estimated in the manner previously described, the calculations proceed in an iterative loop:

Re(Vk)= (Vk*1.689*LOA)/1.27908E-5

Where Re is the Reynolds Number.

Cf= .075/[log(Re)-2]^2

Where Cf is the frictional drag coefficient, calculated by the ITTC convention.

Cr= Cf*[1.5*(D/LOA)^1.5+7*(D/LOA)^3]
+.002*(Cp-.6)

Where Cr is the residual, or form, drag coefficient, calculated after Jackson [4], and the last term is a correction for submarines with considerable parallel mid body.

Ct= Cf+Cr+.00025

Where Ct is the total drag coefficient for the bare hull, and the final term is the correction factor from tow tank studies.

HP(Vk)= (1/PC)*.00872*Vk^3*(Ct*SF+Da)

Where HP(Vk) is the drag in horsepower for the given speed, Vk.

KW(Vk)= .7457*HP(Vk)+HL

Where KW(Vk) is the total load in kilowatts, including non-propulsion loads, for the speed, Vk.

These calculated results for drag in horsepower and total load in kilowatts are displayed on the terminal screen for designer inspection. From these results, the designer may determine the maximum speed of the submarine for the installed shaft horsepower, as well as the submerged endurance at any speed for installed energy storage capacity. If desired, the module may be re-run for various values of propulsive coefficient and non-propulsion loads, until a satisfactory result is obtained.

When the designer accepts the calculated values, the results are written into an output file for hard-copy record purposes.

PRELIMINARY BALANCE

GENERAL DISCUSSION

When the designer has completed the speed and power module, he will proceed to the design of the pressure hull, which is a process covered separately in a concurrent Ocean Engineer thesis by Marvin Meade [11]. With a graphic display of the envelope and pressure hull to serve as a visual aid, the designer will enter the subject module for this chapter to accomplish a longitudinal weight and buoyancy balance.

The purpose of the preliminary balance is to determine the locations of lead and main ballast tanks, in order to confirm the feasibility of the pressure hull design. The required lead (Pb) longitudinal center of gravity (LCG) is achieved by balancing the longitudinal moments of all weights except the main ballast tanks to create an attitude of zero trim in the normal surface condition (NSC).

Following the determination of lead placement, an

additional moment balance is computed to locate the main ballast tank LCG in order to achieve zero trim in the submerged condition.

In addition to the longitudinal balance, a vertical weight balance is computed to provide an estimate of transverse stability while submerged. This estimate is in the form of BG, the distance from the vertical center of buoyancy to the vertical center of gravity of the submarine. This parameter should have a value of about one foot for an optimum design. To simplify calculations, no trim is allowed in the surfaced condition.

CALCULATION PROCESS

The first set of input data required for balancing the submarine is passed from previously completed modules. The data which is read in from earlier calculations is as follows:

All 16 weight categories from the Weight Module (W1-W16)

Longitudinal Centers of Buoyancy at NSC and submerged (LCBNSC and LCBSUB)

Maximum Envelope Radius (R)

Length Overall (LOA).

The first set of keyboard input is the LCG and VCG information for the seven primary weight groups and the

variable load:

LCG(1)-LCG(7)

VCG(1)-VCG(7)

LCG(VL)

VCG(VL)

Although detailed arrangements have not, as yet, been addressed in the design process, the designer must have given enough thought to rough arrangements to provide these location data for the aggregate weight subdivisions. If, when detailed arrangements are completed at some later time, weight group LCG and VCG locations are signifigantly different from the input provided for this module, the balance process must be repeated.

The prog. am now calculates arrays of longitudinal and vertical moment data:

LM(i)= LCG(i)*W(i)

VM(i)= VCG(i)*W(i)

Where LM and VM are the longitudinal and vertical moments of the weight groups, respectively.

$$LCG(Cond.\ A\text{-}1)=\sum_{i=1}^{7} LM(i)\ /\ \sum_{i=1}^{7} W(i)$$

$$VCG(Cond.\ A\text{-}1)=\sum_{i=1}^{7} VM(i)\ /\ \sum_{i=1}^{7} W(i)$$

The designer is now given a screen display of the total weight of lead, and is asked to input the desired

amount of that total to be used as margin lead (ML). This lead will arbitrarily be located at LOA/2 longitudinally and D/2 vertically. The remainder of the lead is considered to be trim lead (TL). The required LCG for trim lead is then calculated:

LCG(TL)= [W(NSC)*LCBNSC -LM(Cond. A-1) -LM(VL) -ML*LOA/2]/SL

Similarly, the actual VCG(Lead):

VCG(Lead)= (TL*6 +ML*R)/W(Lead)

Where 6 feet above the keel is arbitrarily chosen for VCG(TL) to make placement in the circular cross section of the ballast tanks feasible.

The designer is then asked whether the required LCG(SL) is feasible. If the response is affirmative, the program continues on to main ballast tank (MBT) calculations. If not, the designer is asked for new LCG and VCG data for the weight groups and variable load. He is prompted, in each case, with the previously entered LCG and VCG for each category. Of course, if reasonable adjustments in weight group locations cannot lead to feasible stability lead LCG (feasible being, generally, between the physical centers of the main ballast tanks), a re-appraisal of the pressure hull design and rough arrangements is necessary.

When the lead has been satisfactorily located, calculation of required MBT LCG is accomplished (VCG(MBT) is arbitrarily fixed at R):

LCG(NSC)= [LM(Cond. A-1) +LM(Lead)

+LM(VL)]/W(NSC)

LCG(MBT)= [W(Submerged)*LCBSUB

-LCG(NSC)*W(NSC)]/W(MBT)

The designer is again asked to decide whether the calculated required LCG is feasible, with similar results in program flow depending on the answer. Feasibility of MBT location is dependent on the desired geometry of physical locations for MBTs as envisioned by the naval architect. The continued graphic display of envelope and pressure hull geometry is useful in aiding this decision process.

The program now sums all vertical moments for weight groups, lead, variable load and main ballast to arrive at an overall VCG in the submerged condition (VCGSUB). An estimate of submerged stability may be calculated and displayed:

BG= R-VCGSUB

Note that the vertical center of buoyancy is assumed to be at R (D/2) for the submarine, which is basically a body of revolution. Any error due to buoyancy of appendages will be small and conservative, since off-center buoyancy items (sail, etc.) are generally above the geometric center of the envelope. Again, the designer is asked to accept or reject the calculated stability, and, again, rejection results in looping back to the input section of the program module.

When all calculated values are deemed acceptable, the input data and results are written to an output file for record purposes, and the locations of each item's center of gravity are plotted on the pressure hull graphic display. The designer is now ready to proceed to the calculation of the equilibrium polygon.

THE EQUILIBRIUM POLYGON

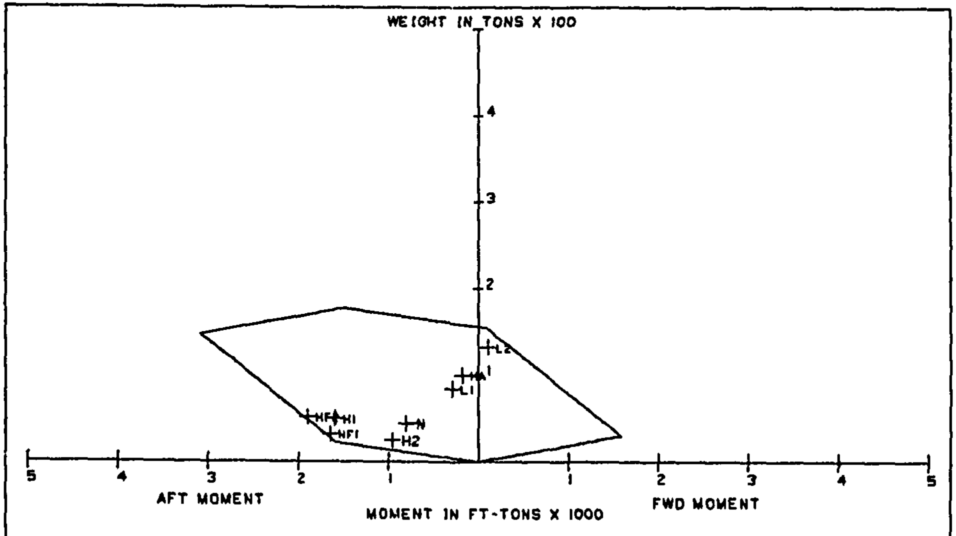
GENERAL DISCUSSION

One of the most important sets of calculations in the submarine design process is the determination of the equilibrium polygon. Any submarine experiences changes in weight distribution during the course of operations, as items in the variable load are consumed. A system of tanks must be sized and located in such a way that this change in weight distribution and amount may be compensated to allow the submarine to remain in proper balance and trim.

A detailed list of the variable load items to be considered may be found later in this chapter. In aggregate, the items compose major groups in the categories of provisions, ammunition, stores, fuel, and other consumable fluids. For a nuclear or other non-fossil-fuel submarine, the boat will always become lighter as items are consumed, and the compensation consists of adding appropriate amounts of variable

ballast at the proper locations. In the case of a diesel or other fossil-fuel submarine, a paradox exists for the designer. As fuel is consumed from the fuel ballast tanks external to the pressure hull, these tanks are compensated with seawater, since they are "soft" tanks, and must remain full of fluid to maintain a zero differential pressure across the tank structure. The seawater used for compensation has a higher density than the consumed fuel which it displaces. Consequently, the submarine actually becomes heavier as fuel is consumed.

The tool which is used to calculate the required locations and capacities of the compensating tanks is called the equilibrium polygon, which is a plot of weight versus moment (fore and aft) of the weight. Figure 7.1 is an example of the equilibrium polygon for a submarine. The points within the polygon represent extreme operating conditions for the submarine in terms of the weight and moment required to balance the conditions. The solid lines forming the "polygon" are plots of the weight and moment added by progressive filling and emptying of the major compensating tank groups. These groups, which may be comprised of several tanks each (the aggregate weight and moment for the group is plotted), are the forward trim, auxiliary, and after trim tank groups. If all of the operating points for the submarine are contained within the polygon, the submarine may be safely compensated for all expected



EQUILIBRIUM POLYGON

FIG. 7.1

operational conditions. If not, the designer must change the location(s), weight(s), or both, for the appropriate group(s), in order to expand the polygon to include all the conditions.

The conditions to be calculated for polygon points are specified in the NAVSHIPS Technical Manual (NSTM), Chapter 9290 [12]. The extreme conditions are more severe than an actual submarine would be expected to encounter, and represent a "worst case" situation for weight and moment balance. These extreme conditions may be thought of as representing the following operational situations:

HEAVY #1: Following a short, fast patrol, during which no ordnance is expended, but all fuel oil is consumed.

HEAVY #2: Same as the previous condition, except that only the fuel oil in the fuel ballast tanks is consumed.

LIGHT #1 & LIGHT #2: Two variations following a short patrol with no fuel consumed, but with all ordnance expended.

HEAVY FORWARD #1: Following a patrol where only aft ordnance is expended, and all fuel oil is consumed

from the forward fuel ballast tanks

HEAVY FORWARD #2: Same as previous condition, except that fuel from normal fuel oil tanks is also consumed.

HEAVY AFT: Following a patrol with only forward ordnance expended, and with fuel oil from the aft fuel ballast and normal fuel oil tanks consumed.

CONDITION N: The "normal" full-load patrol condition at the beginning of a patrol. For submarines with FBTs, the FBTs are specified as fully ballasted with seawater.

CONDITION M: The same as condition N, except that submarines with FBTs carry a full fuel load in the FBTs (for such submarines, this condition will be lighter than condition N).

For the above extreme conditions, water density is also specified within typical ocean ranges, so as to accentuate the worst case nature of the operating conditions.

THE VARIABLE LOAD

The items that comprise the category referred to as the variable load are numerous and diverse in nature. Classification of the items is again aided by the guidelines and categories specified in the NAVSHIPS Technical Manual.

The variable load is subdivided into two large sub-groups; fixed items, not expected to vary significantly during a patrol, and the truly variable items, routinely expended as the submarine operates. The "fixed" portion of the variable load consists of the following items:

CREW AND EFFECTS: Actual weight of the submarine's complement of personnel, clothing, and associated personal effects.

SLBMs OR COMPENSATING WATER: The weight, for a fleet ballistic missile (FBM) submarine, of the submarine launched ballistic missiles (SLBMs) or the compensating water in their absence. SLBM compensating water exactly matches the weight and moment of the missiles themselves, so that this category may, indeed, be considered a fixed item.

SANITARY TANKS: The weight of tanks and contents associated with waste disposal.

OXYGEN CANDLES: The weight of these devices, used for the generation of oxygen under emergency conditions. These are generally a very small weight item.

LUBE OIL IN SUMPS: That weight of lube oil contained in the oil sumps of equipment.

FIXED CLEAN FUEL OIL: That weight of oil carried in non-compensated tanks which are maintained essentially full (i.e., shield tanks on nuclear submarines).

As can be seen from Reference [12], there are other items specified as fixed load, but they are minor in nature, and beyond the level of detail required for conceptual design. For example, the item "depth control tanks" in the NSTM is not included. This omission will be explained in the discussion of fuel load subdivision in the following section.

The items considered variable in nature are as follows:

PROVISIONS AND STORES: The consumable foodstuffs

(provisions) and spare parts, paper supplies, etc. (stores) that the submarine crew will use during a patrol.

REVITALIZATION OXYGEN: Oxygen used for life support (atmosphere control) during lengthy submerged periods. On newer submarines (particularly nuclear), a method of oxygen generation often obviates the need for oxygen banks.

TORPEDOS, MISSILES, AND AMMUNITION: This category is self-explanatory. All expendable ordnance other than SLBMs is included.

WRT TANKS: This item is the "water round torpedo" tank capacity, used to compensate for the presence or absence of torpedos in the tubes.

RESERVE ELECTROLYTE: The electrolyte carried to replenish the submarine storage battery.

FRESH WATER: For this thesis, this is a composite category, consisting of the total of potable, feed, and battery water.

RESERVE LUBE OIL: Lube oil carried to replenish the sump lube oil during the patrol.

FUEL: Propulsion fuel for the submarine. See the discussion that follows for the sub-categories of fuel and their significance to the polygon computations.

Fossil-fuel submarines present a variable load parodox for the designer. Fuel which is carried external to the pressure hull (the majority of the fuel load for most "conventional" submarines) must be replaced by seawater as it is burned, since these external "fuel ballast tanks" (FBTs) are not constructed to withstand a high pressure differential (i.e., they are "soft tanks"). The seawater which is used to compensate these tanks is heavier than the fuel it replaces. Thus, as the submarine consumes fuel, it gets heavier, rather than lighter.

To provide a margin of safety at the base of the equilibrium polygon, some means must be provided to remove weight as fuel is burned. In earlier designs, a "safety tank" was provided, of sufficient capacity that when it was emptied, enough water weight was pumped overboard to compensate for the added seawater in the FBTs as fuel was consumed. The safety tank, then, was full at the beginning of the patrol, and would be empty, or nearly so, when all fuel had been removed from the FBTs. For purposes of this thesis, the safety tank

concept was not used, since the weight and volume of the tank and it's contents is a non-productive load item. However, a means to compensate for the FBT weight addition is necessary.

The method chosen for implementation in this thesis is the provision for a variable fuel oil (VFO) tank. The VFO tank is internal to the pressure hull, and does not require seawater compensation as fuel is consumed. In actual practice, this tank may be ballasted and treated as a dual-purpose tank, acting as an additional trim or auxiliary tank. For computational simplicity within the program module, however, the VFO tank is treated as an uncompensated clean fuel oil tank. An operational mode of employment is assumed whereby the fuel in the VFO tank is consumed first during the patrol, before the FBTs are utilized. Thus, the size of the VFO tank must, at a minimum, be sufficient to compensate for the difference in fuel and seawater densities for the FBT capacity. This dictates a minimum capacity of 23% of the total fuel load. The designer is asked to input the percentage of total fuel to be allocated to the VFO tank (or tank group, as arrangements may dictate). If sufficient internal volume is available, it may be advantageous to use a figure greater than 23%, in order to provide an adequate margin at the base of the polygon.

CALCULATING THE POLYGON

The program module execution begins by reading the weight group values, submerged LCB, and LCG of the main ballast tanks from data files generated by the weight estimating and balance modules. The total load to submerge the submarine is calculated:

Load to Submerge= Submerged Displacement- Condition A Weights.

The total variable load weight is displayed, and the designer is prompted, item by item, for the weight and LCG of each variable load item. As each weight and LCG are entered, the balance of the weight in the variable load account is displayed. Items which are commonly distributed in two sub-items, fore and aft, are input as two separate weights and LCGs. Examples of such items are torpedos, missiles, and fuel. The designer should have given some thought prior to this input sequence to the subdivision of the variable load. The envelope and pressure hull are displayed at the top of the screen during this stage of the module, to facilitate estimating the LCG for each item.

The last item to be entered is fuel. The balance of the variable load account at this point is assumed to be all fuel (obviously, the balance should be zero for a non-fossil-fuel submarine). The total fuel load is

displayed, and the designer is asked to enter the fraction to be allocated to the VFO tank. He is then asked to provide LCGs for the forward and aft FBTs and the VFO tank.

With the input of all items completed, the program displays a variable load summary of all item weights and LCGs. If the summary is satisfactory, the program proceeds with calculations; if not, the designer is returned to the input sequence, where the prompt for each item now includes a display of its current value. If this value is to remain unchanged, the operator simply hits the "RETURN" key; otherwise, a new value may be specified. This iterative input procedure is repeated until the designer is satisfied with the variable load summary.

When the input of all variable load items is complete and accepted by the operator, the program calculates the water to balance, and the associated moment of that water, for each of the nine specified polygon points. Note that conditions M and N will be identical for nuclear submarines. Condition M is only defined for submarines with FBTs.

A computation of one of the equilibrium conditions would proceed in the following manner. Prior to computing the individual conditions, the arms and moments for each of the variable load items are determined:

Item Arm= Submerged LCB -Item LCG

Item Moment= Item Weight *Item Arm

In addition, factors of proportionality are determined for light and heavy seawater. For each condition, the sum of all variable load items, in the fractional amount specified by the NAVSHIPS Technical Manual, is computed. Similarly, the total moment of the specified items is determined. The required water to balance and moment to balance may now be calculated:

Water to Balance= Load to Submerge -MBT -Sum of Item Weights

where MBT= Main Ballast Weight

Moment to Balance= -MBT Moment -Sum of Item Moments

If required, the water and moment to balance are multiplied by the proportionality factor for light or heavy seawater. Each of the nine conditions is calculated in a similar fashion, resulting in arrays of weight values in tons and moments in foot-tons (with negative moment values indicating an aft moment).

Several "housekeeping" chores are performed at this point in the module. The display of envelope and pressure hull, and associated data base attributes, are filed under the part name MIT.HULLOUT, for later recall to plot desired views of this graphic screen. The maximum values of water and moment to balance are examined to determine the required scale for the polygon plot. A new part is activated under the name MIT.POLYOUT

for graphically displaying the polygon. Depending on the scale required, one of several pre-constructed "form" drawings is activated, providing the axes and appropriate scale information for the polygon under consideration.

The program now inserts and labels each of the equilibrium condition points on the graphic display. This gives the designer a clear display of the limits required for his polygon design. An important consideration at this point is whether an adequate margin remains at the base of the polygon. Generally, if the lowest weight value is less than 15 tons from the abscissa, the designer should seriously consider re-structuring the variable load, as necessary, to achieve this margin. This may be accomplished by returning to the input section of the module. An example of such a change would be to place more of the fuel load, volume permitting, in the VFO tank, for a fossil-fuel submarine.

The operator is now asked to provide the LCGs of the Forward Trim, Auxiliary, and After Trim tank groups. He is then prompted for the capacities, in tons, for each of the tank groups. The program determines the arms and moments of each group in the same fashion previously delineated for the variable load items.

Lines are now inserted on the display to represent the successive filling of Forward Trim, Auxiliary, and

After Trim tanks, followed by the successive draining of the tanks in the same order. These lines form the polygon, and all equilibrium points must fit within its boundaries if the tank locations and capacities are adequate. Obviously, a polygon whose boundaries substantially exceed the extent of the equilibrium points, while safe, represents a waste of internal volume, and should not be accepted.

The designer now has the option to accept or reject the polygon, as plotted. If it is rejected, the polygon boundaries are erased, and the designer may specify new tank group capacities, LCGs, or both, after which a new polygon is plotted. This process may be repeated as many times as are required to achieve an acceptable polygon.

When the polygon results are satisfactory, the polygon is plotted on the electrostatic plotter, the part MIT.POLYOUT is deleted, and the variable load summary, tank locations, and tank capacities are written to an output file for a permanent record of the final input data set. One iteration of the submarine conceptual design process, as supported by this thesis, is complete.

SUMMARY AND RECOMMENDATIONS FOR FURTHER RESEARCH

SUMMARY

This thesis constitutes a first exploration of the use of interactive graphics for conceptual design of submarines. The resulting product is useful, flexible, and accurate for conceptual design activity. It is by no means a finished product in the commercial sense, but provides a tool to build on and to use confidently for such applications as student design projects.

The power of real-time graphic display of the design during its creation is a major factor in the worth and potential of the program. Many hours of laborious manual drafting are avoided by the high-resolution graphic output. In addition, a user with even a rudimentary knowledge of the Computervision system can go far beyond the capabilities of the basic software package, thus greatly enhancing its value as a design tool.

The program is highly interactive; the user controls all important design decisions. The designer must have a solid knowledge of naval architecture, which is appropriate for any ship design program. Subjective design judgements must remain in the province of the user for a credible result, and that philosophy has been rigorously adhered to.

RECOMMENDATIONS FOR FURTHER RESEARCH

There are nearly unlimited opportunities for continued work on this project, and in this general area. This thesis has scratched the surface in an area ripe for research and development.

The first obvious areas encompass those portions of the design process which were excluded from the current project. These include structures, propeller design and detailed interior arrangements.

Submarine structural design could benefit greatly from treatment in the CAD environment. Visualization of complex structural relationships is intrinsically clearer with the use of real-time graphic display. In addition, the use of powerful finite element packages, with automatic mesh generation, is possible on many CAD systems, including Computervision. These packages display stress levels in color, and will magnify displacements on command for rapid visual analysis.

Arrangement of interior details is feasible on these systems, including automatic interference checking and layout of piping and electrical distribution networks.

Propeller design may be facilitated by graphic display of the flow regime during analysis. Much analytical work remains in this area, and research into links with main-frame systems and large data bases is also needed.

In addition, the translation of this package into the Fortran-S language, with attendant programming of linking and loading files, would greatly improve execution speed and I/O flexibility. The creation of customized graphics commands would add more power and enhance the ergonomics of the package.

Further work in the subject areas of weight estimation, hydrodynamics, and animated display of control system performance is required to derive the maximum benefit from submarine CAD. All of these subjects have been separately addressed in other software packages, but would be valuable additions to an integrated design program.

With increased interest and design volume in the submarine design field, research funding is both necessary and clearly justified by the potential improvements to the speed and accuracy of the design process. The overall area of computer aided design of

submarines can support major research projects in the future, benefitting both academic and commercial design activities.

BIBLIOGRAPHY

1. CODESUB (Conceptual Design of Submarines) (U). Center for Naval Analyses. Research Report #211 (Confidential), 1973.

2. ASSET. Boeing Computer Services. Project #BCS-40372, 1982.

3. Ship Work Breakdown Structure. Naval Sea Systems Command. NAVSEA 0900-LP-039-9010, 1972.

4. Jackson, Harry. Unpublished Course Notes: M.I.T. Professional Summer Course in Submarine Design Trends, 1983.

5. Summary of Submarine Data (U). Naval Sea Systems Command. PMS 393 (Confidential), 1974.

6. Submarine Poly. U.S. Navy Bureau of Ships.

7. Arentzen, E. and Mandel, Philip. "Naval Architectural Aspects of Submarine Design". Paper for the Annual Meeting. SNAME, 1960.

8. Bukalov, V. and Narusbayev, A. Atomic-Powered Submarine Design. Sudostroyeniy Publishing House (Leningrad), 1964.

9. Newman, J. Marine Hydrodynamics. The M.I.T. Press, 1977.

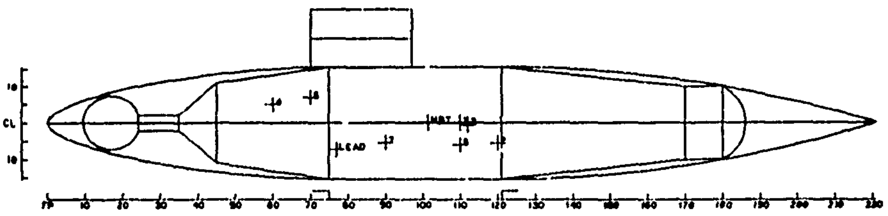
10. Comstock (Ed.). Principles of Naval Architecture. SNAME, 1977.

11. Meade, Marvin. Application of Computer Aided Design to Submarines. O.E. Thesis. Massachusetts Institute of Technology, 1984.

12. Stability and Buoyancy. NAVSHIPS Technical Manual Chapter 9290, 1977.

APPENDIX I:

SAMPLE OUTPUT



PRESSURE HULL SEGMENTS

SEG	FWD	AFT	DFWD	DAFT	VOL	LCB
1	9.5	24.2	0	4	1765.4	17
2	24.2	35	4	4	135.4	29.6
3	35	45	4	22	1533.4	42
4	45	75	22	31	16705.4	61.7
5	75	121	31	31	34719.3	98
6	121	170	31	20	25412.6	142
7	170	180	20	20	3141.6	175
8	180	186.2	20	0	1102.4	182.2
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

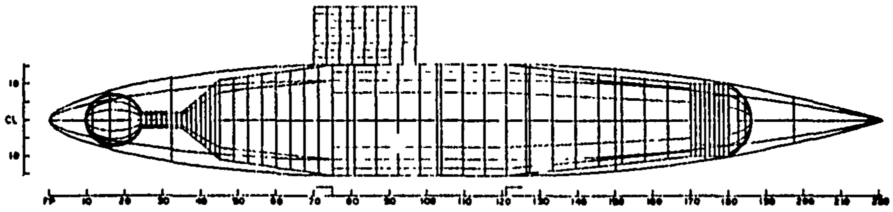
PRESSURE HULL DESIGN SUMMARY

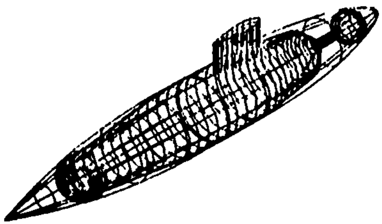
VOLUME REQUIRED:	84521.5
VOLUME USED:	84521.5
VOLUME REMAINING:	0
RESERVED VOLUME	875
CENTER OF BUOYANCY.	105.2

PRINCIPAL CHARACTERISTICS

LENGTH OVERALL:	221
DIAMETER:	31
ENTRANCE ENDS:	75
RUN BEGINS.	121
SUBMERGED LCB:	102.951
SUBMERGED DISP:	2951.5
ENVELOPE DISP.	3099.1

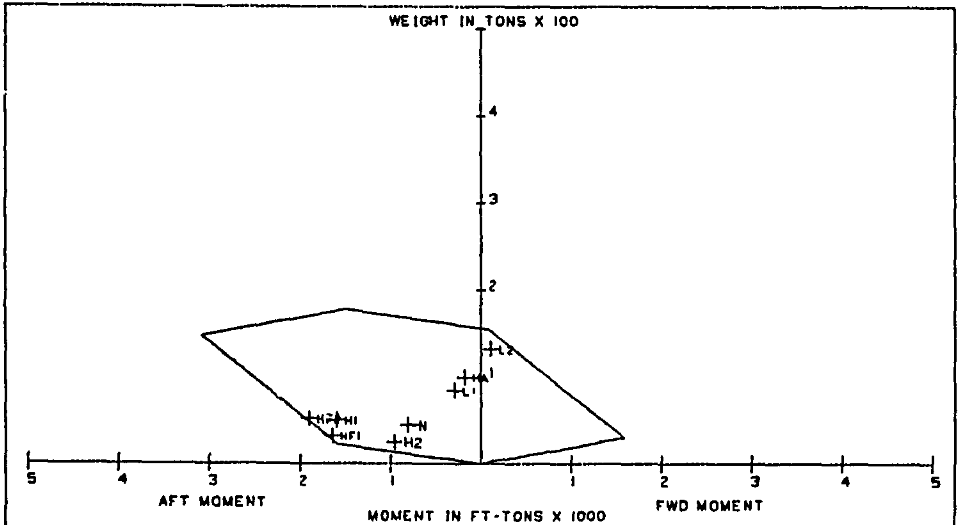
DEMO





PRINCIPAL CHARACTERISTICS	
LENGTH OVERALL:	221
DIAMETER:	31
ENTRANCE ENDS	75
RUN BEGINS:	121
SUBMERGED LCB:	102.951
SUBMERGED DISP.	2951.5
ENVELOPE DISP.	3099.1

DEMO



EQUILIBRIUM POLYGON

```
MIT.&BCD.WEIGHT
 3-28-84 13:37:44 FUTIL  6.18

   1!WEIGHT ESTIMATE FOR 13.461 DESIGN PROJECT:
   2!
   3!TYPE: ATTACK
   4!PROPULSION: MODIFIED WANKEL ELECTRIC
   5!BATTERY TYPE: NICKEL CADMIUM
   6!SPRINT KW-H: 3300
   7!ENDURANCE KW-H: 8000
   8!BATTERY WEIGHT: 165.631
   9!ENGINE HORSEPOWER: 4500
  10!ENGINE & ASSOC. EQUIP. WEIGHT: 22.86
  11!MOTOR & ASSOC. EQUIP. WEIGHT: 32.25
  12!DISCRETE ADDITION TO GROUP 2 WEIGHT: 80
  13!MAX. OPERATING DEPTH IN FEET: 700
  14!
  15!LEAD FRACTION: 0.1
  16!VARIABLE LOAD FRACTION: 0.089
  17!RESERVE BUOYANCY FRACTION: 0.125
  18!FREE FLOOD FRACTION: 0.08
  19!
  20!
  21!CATEGORY      WEIGHT
  22!
  23!GROUP 1      493.318
  24!GROUP 2      300.741
  25!GROUP 3      17.4418
  26!GROUP 4      95
  27!GROUP 5      137.043
  28!GROUP 6      117.3
  29!GROUP 7      85
  30!COND. A1     1245.84
  31!LEAD         124.584
  32!COND. A      1370.43
  33!VAR. LOAD    133.884
  34!NSC          1504.31
  35!MBT          188.039
  36!SUB. DISP.   1692.35
  37!FREE FLD.    135.388
  38!ENV. DISP.   1827.74
```

```
MIT.&BCD.GEOMOUT
 3-28-84 13:38:56 FUTIL  6.18

   1!GEOMETRY OUTPUT FOR 13.461 DESIGN PROJECT:
   2!
   3!
   4!LENGTH OVERALL: 139.947
   5!MID-BODY LENGTH: 74.9465
   6!OVERALL PRISMATIC: 0.798368
   7!LENGTH/DIAMETER: 5.1832
   8!SUBMERGED LCB (AFT OF FP): 66.2489
   9!DRAFT @ NORMAL SURF. COND.: 21.9585
  10!LCB @ NORMAL SURF. COND.  : 66.3329
  11!
  12!RECAP OF INPUT DATA:
  13!
  14!ENVELOPE DISPLACEMENT :1827.74
  15!ENTRANCE: 25
  16!RUN     : 40
  17!DIAMETER: 27
  18!FWD PRISMATIC: 0.6
  19!AFT PRISMATIC: 0.55
```

MIT.&BCD.OFFOUT
3-28-84 13:38:56 FUTIL 6.18

!!LENGTH WRT FP	OFFSET
2!0	0
3!0.5	1.79298
4!1.5	3.41525
5!2.99946	5.08919
6!4.99839	6.77129
7!6.99732	8.1157
8!13.9946	11.3674
9!20.992	13.1284
10!27.9893	13.5
11!34.9866	13.5
12!41.984	13.5
13!48.9813	13.5
14!55.9786	13.5
15!62.9759	13.5
16!69.9733	13.5
17!76.9706	13.5
18!83.9679	13.5
19!90.9652	13.5
20!97.9625	13.5
21!104.96	13.334
22!111.957	12.444
23!118.955	10.7078
24!125.952	8.07667
25!132.949	4.51632
26!139.947	0

```
MIT.&BCD.POWER
 4- 9-84 10:13:59 FUTIL  6.18

  1!SPEED AND POWER RESULTS FOR 13.461 DESIGN PROJECT:
  2!
  3!NON-PROPULSION LOADS (KW): 135
  4!PROPULSIVE COEFFICIENT    : 0.77
  5!
  6!
  7!SPEED (KT)      SHP          TOTAL KW
  8!
  9! 1             0.615861      135.459
 10! 2             4.61783       138.444
 11! 3             15.0496       146.222
 12! 4             34.8408       160.981
 13! 5             66.8572       184.855
 14! 6             113.92        219.95
 15! 7             178.818       268.345
 16! 8             264.312       332.097
 17! 9             373.141       413.251
 18! 10             508.023       513.833
 19! 11             671.663       635.859
 20! 12             866.75        781.335
 21! 13             1095.96       952.256
 22! 14             1361.95       1150.61
 23! 15             1667.38       1378.37
 24! 16             2014.9        1637.51
 25! 17             2407.13       1930
 26! 18             2846.71       2257.79
 27! 19             3336.24       2622.83
 28! 20             3878.34       3027.08
 29! 21             4475.63       3472.47
 30! 22             5130.68       3960.95
 31! 23             5846.11       4494.44
 32! 24             6624.47       5074.87
 33! 25             7468.37       5704.17
 34! 26             8380.38       6384.25
 35! 27             9363.07       7117.04
 36! 28             10419         7904.45
 37! 29             11550.7       8748.37
 38! 30             12760.8       9650.74
 39! 31             14051.8       10613.4
 40! 32             15426.3       11638.4
 41! 33             16886.8       12727.5
 42! 34             18435.8       13882.6
 43! 35             20075.9       15105.6
 44! 36             21809.6       16398.4
 45! 37             23639.5       17762.9
 46! 38             25568         19201.1
 47! 39             27597.7       20714.6
 48! 40             29731.2       22305.5
```

```
MIT.&BCD.BALOUT
 5- 6-84 10:54:37 FUTIL 6.21

RESULTS FROM BALANCE MODULE

GROUP  WEIGHT     LCG      VCG

 1     860.354    110      15.5
 2     623.431    120      10
 3     99.814     112      15
 4     105        60       20.5
 5     243.99     110      9.5
 6     195.5      70       22.5
 7     90         90       10
 A1    2218.09    106.19   13.902
 LEAD  221.809    76.889   8.1414
 A     2439.9     103.53   13.378
 V LD  183.648    98       16
 MBT   327.943    101.39   15.5

MARGIN LEAD (TONS): 50 @VCG=D/2. LCG=LOA/2
STABILITY LEAD (TONS): 171.809 @VCG= 6 FT
STABILITY LEAD REQUIRED LCG: 67.1016

NSC (TONS): 2623.55 @LCB: 103.145

SUBM (TONS): 2951.49 @LCB: 102.951

SUBMERGED STABILITY (BG): 1.72272 FT
```

```
MIT.OUT.&BCD.POLY
 5- 7-84 19:24:02 FUTIL 6.21

POLYGON SUMMARY OUTPUT FILE


    ITEM                          LCG              WEIGHT

CREW AND EFFECTS              55           6
SLBM'S OR COMP. WATER         0            0
SANITARY TANKS                85           1.5
LUBE OIL IN SUMPS             90           5
FIXED CLEAN FUEL OIL          0            0
****TOTAL FIXED ITEMS****     78.913            11.5
PROVISIONS AND STORES         75           9.5
REVITALIZATION O2             88           1
TORPEDOS IN FWD ROOM          45           27
TORPEDOS IN AFT ROOM          0            0
TACTICAL MISSILES FWD         0            0
TACTICAL MISSILES AFT         116          13
WRT TANKS                     30           5
RESERVE ELECTROLYTE           58           1
TOTAL FRESH WATER             100          11
RESERVE LUBE OIL              95           5.5
FUEL BALLAST FWD              26.5          28
FUEL BALLAST AFT              115           21.074
VARIABLE FUEL OIL             45           49.074


VARIABLE BALLAST TANK DATA:
   TANK GROUP                   LCG              CAPACITY (TONS)
FORWARD TRIM                    35                25
AUXILIARY                       75                100
AFTER TRIM                      105               35
```

APPENDIX II:

SOURCE CODE LISTINGS

MIT.4BCD.DESIGN
5- 7-84 13:50:00 FUTIL 6.21

```
DIM W(17),P(17),F(8),RS(21),CIRC(21),LX(8),RX(8),XPOS(30),SA(21)<#
>,LS(21),HP(41),KW(41),DU(21),X(29),Y(29)
READ (ENTER DESIGN NAME: ) ANAME
#WEIGHT CONTINUE
N=1
REPEAT
    N=N+1
    IF (N.EQ.3.OR.N.EQ.5) GOTO LOOP
    PRNT DO YOU WANT TO SPECIFY (S) OR CALCULATE (C) GROUP [N] WEIGHTS?
    READ &R
    IF (&R.EQ."S") F(N)=1
    IF (&R.EQ."C") F(N)=0
    IF (F(N).EQ.0) GOTO LOOP
    PRNT ENTER GROUP [N] WEIGHT (TONS).
    READ WT
    W(N)=WT
#LOOP CONTINUE
UNTIL(N.EQ.7)
READ(MAX. OPERATING DEPTH IN FEET: )ZMAX
P(1)=.2155+1.606225E-4*ZMAX
READ(RESERVE BUOYANCY FRACTION: )P(13)
READ(LEAD FRACTION: )P(9)
READ(VARIABLE LOAD FRACTION: )P(11)
READ(FREE FLOOD FRACTION: )P(15)
PRNT NUCLEAR (1), DIESEL (2), OR WANKEL (3) PROPULSION?
READ PM
PRNT ATTACK (1) OR FBM (2)?
READ T
IF (F(2).EQ.1) GOTO LAB1
GOSUB GROUP2
#LAB1 IF (F(6).EQ.1) GOTO A1
GOSUB GROUP6
GOTO A1
#GROUP2
    READ (SHAFT HORSEPOWER: )SHP
    IF (PM.NE.1) GOTO BATT
    K=.03*SHP/1000+1.13
    W1=(K*1E5*SHP)/(LG(SHP))**5
    W(2)=W1/2240
RTNSUB
#BATT CONTINUE
    PRNT BATTERY TYPE: NICAD (1), VARTA (2), OR TRIDENT (3)?
    READ BT
    READ(TOTAL ENGINE HORSEPOWER: )ESP
    IF (PM.EQ.2) KE=.00968
    IF (PM.EQ.3) KE=.00508
    WE=KE*ESP
    WW=.00645*SHP
    READ(TOTAL KW-H STORAGE FOR SPRINT: )KSP
    READ(TOTAL KW-H STORAGE FOR ENDURANCE RATE: )KEND
    IF (BT.EQ.1) CS=35.5616
    IF (BT.EQ.2) CS=22.353
    IF (BT.EQ.3) CS=12.1925
    B=KSP/CS
    IF (BT.EQ.1) CE=48.3
    IF (BT.EQ.2) CE=42.674
```

```
    IF (BT.EQ.3) CE=25.4012
    BI=KEND/CE
    IF (BI.GT.B) B=BI
    WT=WE+WM+B
    PRNT ENTER DISCRETE ADJUSTMENT TO GROUP 2 (O OR # OF TONS).
    READ DA
    W(2)=WT+DA
RTNSUB
#GROUP6
   READ(NUMBER OF PERSONNEL: ) NP
   W(6)=2.3*NP
RTNSUB
#AI
IF (PM.EQ.1) P(3)=.045
IF (PM.EQ.2.OR.PM.EQ.3) P(3)=.014
P(4)=.06
IF (T.EQ.1) P(5)=.11
IF (T.EQ.2) P(5)=.085
IF (T.EQ.1.AND.PM.EQ.1) P(7)=.03
IF (T.EQ.1.AND.PM.EQ.2) P(7)=.04
IF (T.EQ.1.AND.PM.EQ.3) P(7)=.05
IF (T.EQ.2) P(7)=.14
FW=(P(1)+P(1)*P(9))/(1-P(11))+P(3)+P(5)
IF (F(4).EQ.0.AND.F(7).EQ.0) GOTO CASE1
IF (F(4).EQ.1.AND.F(7).EQ.0) GOTO CASE2
IF (F(4).EQ.0.AND.F(7).EQ.1) GOTO CASE3
IF (F(4).EQ.1.AND.F(7).EQ.1) GOTO CASE4
#CASE1
    AI=(W(2)+W(6))/(1-FW-P(4)-P(7))
    W(4)=P(4)*AI
    W(7)=P(7)*AI
GOTO TOTAL
#CASE2
   AI=(W(2)+W(4)+W(6))/(1-FW-P(7))
   W(7)=P(7)*AI
GOTO TOTAL
#CASE3
    AI=(W(2)+W(6)+W(7))/(1-FW-P(4))
    W(4)=P(4)*AI
GOTO TOTAL
#CASE4
    AI=(W(2)+W(4)+W(6)+W(7))/(1-FW)
#TOTAL
W(3)=P(3)*AI
W(5)=P(5)*AI
W(8)=AI
W(9)=P(9)*AI
W(10)=AI+W(9)
W(12)=W(10)/(1-P(11))
W(11)=P(11)*W(12)
W(1)=P(1)*W(12)
W(13)=P(13)*W(12)
W(14)=W(12)+W(13)
W(15)=P(15)*W(14)
W(16)=W(15)+W(14)
PRNT CALCULATIONS COMPLETE...
PRNT
PRNT
<# OUTPUT SECTION
IF (T.EQ.1) $T="ATTACK"
```

```
IF (T.EQ.2) &T="BALLISTIC MISSILE"
IF (PM.EQ.1) &PM="NUCLEAR"
IF (PM.EQ.2) &PM="DIESEL ELECTRIC"
IF (PM.EQ.3) &PM="WANKEL ELECTRIC"
IF (DA.NE.0) &M="MODIFIED "
IF (DA.EQ.0) &M=""
IF (BT.EQ.1) &BT="NICKEL CADMIUM"
IF (BT.EQ.2) &BT="LEAD ACID (VARTA)"
IF (BT.EQ.3) &BT="LEAD ACID (TRIDENT)"
PRNT WEIGHT ESTIMATE FOR [&NAME]:
PRNT
PRNT TYPE: [&T]
PRNT PROPULSION: [&M][&PM]
IF (PM.EQ.1) GOTO SHAFT
PRNT BATTERY TYPE: [&BT]
PRNT SPRINT KW-H: [KSP]
PRNT ENDURANCE KW-H: [KEND]
PRNT BATTERY WEIGHT: [B]
PRNT ENGINE HORSEPOWER: [ESP]
PRNT ENGINE WEIGHT: [WE]
PRNT MOTOR WEIGHT: [WM]
IF (DA.NE.0) PRNT DISCRETE ADDITION TO GROUP 2: [DA]
*SHAFT CONTINUE
PRNT SHAFT HORSEPOWER: [SHP]
PRNT MAX. OPERATING DEPTH (FT): [ZMAX]
IF (F(6).EQ.1) PRNT NUMBER OF PERSONNEL: [NP]
PRNT LEAD FRACTION: [P(9)]
PRNT VARIABLE LOAD FRACTION: [P(11)]
PRNT RESERVE BUOYANCY FRACTION: [P(13)]
PRNT FREE FLOOD FRACTION: [P(15)]
PRNT HIT <RETURN> TO CONTINUE...
READ &CONT
PRNT
PRNT CATEGORY     WEIGHT
PRNT
I=1
REPEAT
PRNT GROUP [I]   [W(I)]
I=I+1
UNTIL (I.EQ.8)
PRNT COND. AI     [W(8)]
PRNT LEAD         [W(9)]
PRNT COND. A      [W(10)]
PRNT VAR. LOAD    [W(11)]
PRNT NSC          [W(12)]
PRNT MBT          [W(13)]
PRNT SUB. DISP.   [W(14)]
PRNT FREE FLD.    [W(15)]
PRNT ENV. DISP.   [W(16)]
PRNT
PRNT SATISFACTORY (S) OR RECALCULATE (R)?
READ &DECID
IF (&DECID.EQ."R") GOTO WEIGHT
PRNT PLEASE WAIT WHILE OUTPUT AND PASS FILES ARE WRITTEN. THE OUTPUT FOR
PRNT THE WEIGHT CALCULATIONS WILL BE IN FILE "WIT.WEIGHT". WHEN YOU HAVE
PRNT COMPLETED THE ENTIRE DESIGN PROGRAM, "WIT.WEIGHT" AND THE OUTPUT
PRNT FILES FOR SUBSEQUENT MODULES MAY BE OBTAINED IN HARD COPY BY USING
PRNT THE COMMAND 'PRINTLP"FILENAME"'.
OPENW 2,"WIT.WEIGHT"
&P="  "
```

```
177!&SHP=SHP
178!&ESP=ESP
179!&KSP=KSP
180!&KEND=KEND
181!&ZMAX=ZMAX
182!&NP=NP
183!&BB=B
184!&WM=WM
185!&WE=WE
186!&DA=DA
187!&LF=P(9)
188!&VLF=P(11)
189!&RBF=P(13)
190!&FFF=P(15)
191!&W1=W(1)
192!&W2=W(2)
193!&W3=W(3)
194!&W4=W(4)
195!&W5=W(5)
196!&W6=W(6)
197!&W7=W(7)
198!&W8=W(8)
199!&W9=W(9)
200!&W10=W(10)
201!&W11=W(11)
202!&W12=W(12)
203!&W13=W(13)
204!&W14=W(14)
205!&W15=W(15)
206!&W16=W(16)
207!&X="WEIGHT ESTIMATE FOR "+&NAME+":"
208!GOSUB 0
209!WRITEF 2,&P
210!&X="TYPE: "+&T
211!GOSUB 0
212!&X="PROPULSION: "+&M+&PM
213!IF (PM.EQ.1) GOTO NUKE
214!GOSUB 0
215!&X="BATTERY TYPE: "+&BT
216!GOSUB 0
217!&X="SPRINT KW-H: "+&KSP
218!GOSUB 0
219!&X="ENDURANCE KW*H: "+&KEND
220!GOSUB 0
221!&X="BATTERY WEIGHT: "+&BB
222!GOSUB 0
223!&X="ENGINE HORSEPOWER: "+&ESP
224!GOSUB 0
225!&X="ENGINE & ASSOC. EQUIP. WEIGHT: "+&WE
226!GOSUB 0
227!&X="MOTOR & ASSOC. EQUIP. WEIGHT: "+&WM
228!GOSUB 0
229!IF (DA.EQ.0) GOTO NUKE
230!&X="DISCRETF ADDITION TO GROUP 2 WEIGHT: "+&DA
231!GOSUB 0
232!*NUKE CONTINUE
233!&X="MAX. OPERATING DEPTH IN FEET: "+&ZMAX
234!GOSUB 0
235!WRITEF 2,&P
236!&X="LEAD FRACTION: "+&LF
```

```
237!GOSUB 0
238!&X="VARIABLE LOAD FRACTION: "+&VLF
239!GOSUB 0
240!&X="RESERVE BUOYANCY FRACTION: "+&RBF
241!GOSUB 0
242!&X="FREE FLOOD FRACTION: "+&FFF
243!GOSUB 0
244!WRITEF 2.&P
245!WRITEF 2.&P
246!&X="CATEGORY     WEIGHT"
247!GOSUB 0
248!WRITEF 2.&P
249!&X="GROUP 1      "+&W1
250!GOSUB 0
251!&X="GROUP 2      "+&W2
252!GOSUB 0
253!&X="GROUP 3      "+&W3
254!GOSUB 0
255!&X="GROUP 4      "+&W4
256!GOSUB 0
257!&X="GROUP 5      "+&W5
258!GOSUB 0
259!&X="GROUP 6      "+&W6
260!GOSUB 0
261!&X="GROUP 7      "+&W7
262!GOSUB 0
263!&X="COND. A1     "+&W8
264!GOSUB 0
265!&X="LEAD         "+&W9
266!GOSUB 0
267!&X="COND. A      "+&W10
268!GOSUB 0
269!&X="VAR. LOAD    "+&W11
270!GOSUB 0
271!&X="NSC          "+&W12
272!GOSUB 0
273!&X="MBT          "+&W13
274!GOSUB 0
275!&X="SUB. DISP.   "+&W14
276!GOSUB 0
277!&X="FREE FLD.    "+&W15
278!GOSUB 0
279!&X="ENV. DISP.   "+&W16
280!GOSUB 0
281!GOTO DONE
282!#0 CONTINUE
283!WRITEF 2.&X
284!RTNSUB
285!#DONE CONTINUE
286!OPENW 3."MIT.PASS,"
287!WRITEF 3.&W12
288!WRITEF 3.&W15
289!WRITEF 3.&W16
290!WRITEF 3.&NAME
291!OPENW 4."MIT.HALPASS"
292!WRITEF 4. &W1
293!WRITEF 4. &W2
294!WRITEF 4. &W3
295!WRITEF 4. &W4
296!WRITEF 4. &W5
```

```
297!WRITEF 4. &W6
298!WRITEF 4. &W7
299!WRITEF 4. &W8
300!WRITEF 4. &W9
301!WRITEF 4. &W10
302!WRITEF 4. &W11
303!WRITEF 4. &W12
304!WRITEF 4. &W13
305!WRITEF 4. &W14
306!WRITEF 4. &W15
307!WRITEF 4. &W16
308!OPENW 5."MIT.PASSPH2"
309!WRITEF 5.&W10
310!WRITEF 5.&NAME
311!RUN NEW MIT.G3
312!END
```

MIT.&BCD.C3
4-27-84 11:08:26 FUTIL 6.21

```
DIM RS(21),CIRC(21),LX(8),RX(8),XPOS(30),SA(21),LS(21)<#
>,HF(41),KW(41),DU(21),X(132),Y(132),SX(21)
#GEOM
OPENR 1,"MIT.PASS1"
READF 1,&W
DNSC=&W
READF 1,&W
DFF=&W
READF 1,&W
DENV=&W
READF 1,&NAME
PI=3.14159265
#BRANCH CONTINUE
READ(ENTER LENGTH OF ENTRANCE: )LF
READ(ENTER LENGTH OF RUN: )LA
READ(ENTER HULL DIAMETER: )D
READ(ENTER FOREBODY PRISMATIC COEFF.: )CPF
READ(ENTER AFTERBODY PRISMATIC COEFF.: )CPA
L=LF+LA
LL=LF
INTV=L/20
R=D/2
NF=71.0477*CPF**5+65.8107*CPF**4-483.3037*CPF**3+587.4137*CPF**2<#
>-281.7224*CPF+49.5876
NA=379.6546*CPA**5-938.4708*CPA**4+944.8853*CPA**3-471.0872*CPA**2<#
>+119.1465*CPA-11.3454
LS(1)=0
LS(21)=L
J=2
REPEAT
LS(J)=(J-1)*INTV
XL=LS(J)
GOSUB RAD
RS(J)=RANS
SA(J)=PI*RS(J)**2
DU(J)=SA(J)
J=J+1
UNTIL(J.EQ.21)
GOSUB INTO
VENV=35*DENV
DV=VENV-ANS
LMB=DV/(PI*R**2)
LL=LF+LMB
LOA=LF+LMB+LA
INTV=LOA/20
LS(21)=LOA
LS(1)=0
J=2
REPEAT
LS(J)=(J-1)*INTV
XL=LS(J)
GOSUB RAD
RS(J)=RANS
CIRC(J)=2*PI*RS(J)
SA(J)=PI*RS(J)**2
DU(1)=CIRC(J)
```

```
57!J=J+1
58!UNTIL(J.EQ.21)
59!GOSUB INTG
60!ST=ANS
61!VCYL=PI*LOA*R**2
62!CP=VENV/VCYL
63!J=2
64!REPEAT
65!DU(J)=LS(J)*SA(J)
66!J=J+1
67!UNTIL(J.EQ.21)
68!GOSUB INTG
69!LCBSUB=ANS/VENV
70!DR=.5*R
71!DN=DNSC+.8*DFF
72!LO=DN-.005*DN
73!HI=DN+.005*DN
74!PRNT DN: [DN]
75!PRNT LO: [LO]
76!PRNT HI: [HI]
77!#LCBLP CONTINUE
78!PRNT ITERATING ON NSC DRAFT # T= [R+DR]
79!J=2
80!REPEAT
81!IF(DR.GE.RS(J)) GOTO SAME
82!AC=(ACOS(DR/RS(J)))*(PI/180)
83!SX(J)=SA(J)-(RS(J)**2*AC-DR*(RS(J)**2-DR**2)**.5)
84!GOTO SKIP
85!#SAME CONTINUE
86!SX(J)=SA(J)
87!#SKIP CONTINUE
88!DU(J)=SX(J)
89!J=J+1
90!UNTIL(J.EQ.21)
91!GOSUB INTG
92!DM=ANS/35
93!IF(DM.GE.LO.AND.DM.LE.HI) GOTO MOM
94!IF(DM.GT.HI) GOTO MINUS
95!DR=DR+.1*DR
96!GOTO LCBLP
97!#MINUS CONTINUE
98!DR=DR-.1*DR
99!GOTO LCBLP
100!#MOM CONTINUE
101!J=2
102!REPEAT
103!DU(J)=LS(J)*SX(J)
104!J=J+1
105!UNTIL(J.EQ.21)
106!GOSUB INTG
107!LCBNSC=ANS/(DM*35)
108!DX=(INTV-2)/5
109!LX(1)=.5
110!LX(2)=1
111!LX(3)=1.5
112!LX(4)=2
113!REPEAT LOOP2:J=5,1,(J.GT.8)
114!LX(J)=(J-4)*DX+2
115!#LOOP2 CONTINUE
116!REPEAT EXTRA:I=1,1,(I.GT.8)
```

```
XL=LX(J)
GOSUB RAD
RX(J)=RANS
#EXTRA
GOTO GOUT
#RAD CONTINUE
IF(XL.LE.LF) GOTO FRONT
IF(XL.GT.LF.AND.XL.LE.LL) RANS=R
IF(XL.GT.LL) GOTO AFT
GOTO LOOP4
#FRONT CONTINUE
XX=LF-XL
RANS=R*((1-(XX/LF)**NF)**(1/NF))
GOTO LOOP4
#AFT CONTINUE
XX=XL-LL
RANS=R*(1-(XX/LA)**NA)
#LOOP4 RTNSUB
#INTG <#SIMPSON'S RULE INTEGRATION
ODD=0
REPEAT ODDL:J=2,2,(J.GT.20)
ODD=ODD+4*DU(J)
#ODDL CONTINUE
EVEN=0
REPEAT EVENL:J=3,2,(J.GT.19)
EVEN=EVEN+2*DU(J)
#EVENL CONTINUE
ANS=(INTV/3)*(DU(1)+ODD+EVEN+DU(21))
RTNSUB
#GOUT
LD=LOA/D
PRNT
PRNT
PRNT ENVELOPE DISPLACEMENT: [DENV]
PRNT LENGTH OVERALL: [LOA]
PRNT OVERALL PRISMATIC: [CP]
PRNT DIAMETER: [D]
PRNT LENGTH/DIAMETER: [LD]
PRNT MIDBODY LENGTH: [LMB]
PRNT FWD PRISMATIC: [CPF]
PRNT AFT PRISMATIC: [CPA]
PRNT ENTRANCE: [LF]
PRNT RUN: [LA]
PRNT LCB SUBMERGED: [LCBSUB]
PRNT LCB @ NSC     : [LCBNSC]
PRNT DRAFT @ NSC   : [R+DR]
PRNT SATISFACTORY (S) OR RECALCULATE (R)?
READ &DECID
IF (&DECID.EQ."R") GOTO BRANCH
PRNT PLEASE WAIT WHILE OUTPUT AND PASS FILES ARE WRITTEN. THE OUTPUT
PRNT FILE FOR GEOMETRY SUMMARY DATA WILL BE "MIT.GEOMOUT". A FILE OF
PRNT OFFSETS WILL BE IN "MIT.OFFOUT".
OPENW 3,"MIT.OFFOUT"
&Q="LENGTH WRT FP                    OFFSET"
GOSUB W
LS(1)=0
RS(1)=0
&X=LS(1)
&C=RS(1)
&S="                    "
```

```
&O=&X+&S+&C
WRITEF 3,&O
REPEAT LOOP5:J=1,1,(J.GT.8)
&X=LX(J)
&C=RX(J)
&O=&X+&S+&C
GOSUB W
J=J+1
#LOOP5 CONTINUE
REPEAT LOOP6:J=2,1,(J.GT.21)
&X=LS(J)
&C=RS(J)
&O=&X+&S+&C
GOSUB W
#LOOP6 CONTINUE
GOTO FILE
#W CONTINUE
WRITEF 3,&O
RTNSUB
#FILE CONTINUE
OPENW 4,"MIT.PASSHP"
&X=CP
WRITEF 4,&X
&X=LOA
WRITEF 4,&X
&X=D
WRITEF 4,&X
&X=SF
WRITEF 4,&X
WRITEF 4,&NAME
OPENW 5,"MIT.GEOMOUT"
&X="GEOMETRY OUTPUT FOR "+&NAME+":"
WRITEF 5,&X
WRITEF 5,&S
WRITEF 5,&S
&P=LOA
&X="LENGTH OVERALL: "+&P
WRITEF 5,&X
&P=LMB
&X="MID-BODY LENGTH: "+&P
WRITEF 5,&X
&P=CP
&X="OVERALL PRISMATIC: "+&P
WRITEF 5,&X
&P=LD
&X="LENGTH/DIAMETER: "+&P
WRITEF 5,&X
&P=LCBSUB
&X="SUBMERGED LCB (AFT OF FP): "+&P
WRITEF 5,&X
TT=R+DR
&P=TT
&X="DRAFT @ NORMAL SURF. COND.: "+&P
WRITEF 5,&X
&P=LCBNSC
&X="LCB @ NORMAL SURF. COND.   : "+&P
WRITEF 5,&X
WRITEF 5,&S
&X="RECAP OF INPUT DATA:"
WRITEF 5,&X
```

```
WRITEF 5,&S
&P=DENV
&X="ENVELOPE DISPLACEMENT :"+&P
WRITEF 5,&X
&P=LF
&X="ENTRANCE: "+&P
WRITEF 5,&X
&P=LA
&X="RUN      : "+&P
WRITEF 5,&X
&P=D
&X="DIAMETER: "+&P
WRITEF 5,&X
&P=CPF
&X="FWD PRISMATIC: "+&P
WRITEF 5,&X
&P=CPA
&X="AFT PRISMATIC: "+&P
WRITEF 5,&X
OPENW 6,"MIT.LCB"
&X=LCBNSC
WRITEF 6,&X
&X=LCBSUB
WRITEF 6,&X
&X=R
WRITEF 6,&X
&X=LOA
WRITEF 6,&X
OPENW 7,"MIT.PASSPH1"
&X=D
WRITEF 7,&X
&X=LOA
WRITEF 7,&X
&X=LF
WRITEF 7,&X
&X=LA
WRITEF 7,&X
&X=LCBSUB
WRITEF 7,&X
DSUB=DENV-DFF
&X=DSUB
WRITEF 7,&X
&X=DENV
WRITEF 7,&X
&X=NF
WRITEF 7,&X
&X=NA
WRITEF 7,&X
#DRAW
X(1)=0
Y(1)=0
I=2
REPEAT
X(I)=LX(I-1)
Y(I)=RX(I-1)
I=I+1
UNTIL(I.GT.9)
LFINT=(LF-(INTV+1))/23
XL=INTV+1
I=10
```

```
297!REPEAT
298!X(I)=XL
299!GOSUB RAD
300!Y(I)=RANS
301!XL=XL+LFINT
302!I=I+1
303!UNTIL(I.GT.31)
304!LINT=(LOA-LF)/101
305!XL=LF+LINT
306!I=32
307!REPEAT
308!X(I)=XL
309!GOSUB RAD
310!Y(I)=RANS
311!XL=XL+LINT
312!I=I+1
313!UNTIL(I.GT.131)
314!X(132)=LOA
315!Y(132)=0
316!OPENW 8,"MIT.POINTS"
317!J=1
318!REPEAT
319!&X=X(J)
320!&Y=Y(J)
321!WRITEF 8,&X
322!WRITEF 8,&Y
323!J=J+1
324!UNTIL(J.GT.132)
325!RUN NEW MIT.SPEED
326!END
```

MIT.&BCD.SPEED
4-11-84 13:20:34 FUTIL 6.18

```
DIM HP(40),KW(40)
READ(ENTER HOTEL AND COMBAT SYSTEM LOADS IN KW: )HL
READ(ENTER PROPULSIVE COEFFICIENT: )PC
OPENR 1,"MIT.PASSHP"
READF 1,&CP
READF 1,&LOA
READF 1,&D
READF 1,&SF
READF 1,&NAME
CP=&CP
LOA=&LOA
D=&D
SF=&SF
DA=1.09065E-3*LOA*D+11.25
J=1
REPEAT
RE=(J*1.689*LOA)/1.27908E-5
CF=.075/(LG(RE)-2)**2
CR=CF*(1.5*(D/LOA)**1.5+7*(D/LOA)**3)+.002*(CP-.6)
CT=CF+CR+.00025
HP(J)=(1/PC)*.00872*J**3*(CT*SF+DA)
KW(J)=.7457*HP(J)+HL
J=J+1
UNTIL(J.GT.40)
PRNT
PRNT
PRNT
PRNT SPEED AND POWER OUTPUT FOR [&NAME]
PRNT
PRNT NON PROPULSION LOADS (KW): [HL]
PRNT PROPULSIVE COEFFICIENT   : [PC]
PRNT
PRNT HIT <RETURN> TO  SEE OUTPUT VALUES...
READ &CONT
PRNT
PRNT SPEED (KT)     SHP        TOTAL KW
PRNT
J=1
REPEAT
PRNT [J]            [HP(J)]      [KW(J)]
J=J+1
UNTIL(J.GT.20)
PRNT HIT <RETURN> TO CONTINUE...
READ &CONT
PRNT SPEED (KT)     SHP        TOTAL KW
PRNT
REPEAT
PRNT [J]            [HP(J)]      [KW(J)]
J=J+1
UNTIL(J.GT.40)
PRNT PLEASE WAIT WHILE THE OUTPUT FILE IS CREATED. OUTPUT FOR THIS
PRNT MODULE WILL BE IN FILE "MIT.POWER".
OPENW 2,"MIT.POWER"
&X="SPEED AND POWER RESULTS FOR "+&NAME+":"
WRITEF 2,&X
&PAD1="       "
```

```
&PAD2="     "
WRITEF 2,&PAD1
&HL=HL
&PC=PC
&X="NON-PROPULSION LOADS (KW): "+&HL
WRITEF 2,&X
&X="PROPULSIVE COEFFICIENT    : "+&PC
WRITEF 2,&X
WRITEF 2,&PAD1
&X="SPEED (KT)      SHP         TOTAL KW"
WRITEF 2,&PAD1
WRITEF 2,&X
WRITEF 2,&PAD1
J=1
REPEAT
&J=J
&X=HP(J)
&HP=&X(1,8)
&Y=KW(J)
&KW=&Y(1,8)
&OUT=" "+&J+&PAD1+"    "+&HP+&PAD2+&KW
WRITEF 2,&OUT
J=J+1
UNTIL(J.GT.40)
RUN NEW MIT.ENV
END
```

MIT.&BCD.BALANCE
4-27-84 10:47:27 FUTIL 6.21

```
DIM W(16),LC(16),VC(16),LM(16),VM(16),VY(16)
OPENR 1,"MIT.BALPASS"
J=1
REPEAT
READF 1,&X
W(J)=&X
J=J+1
UNTIL(J.GT.16)
OPENR 2,"MIT.LCB"
READF 2,&X
LCBNSC=&X
READF 2,&X
LCBSUB=&X
READF 2,&X
R=&X
READF 2,&X
LOA=&X
PRNT THE INPUT DATA FOR THIS MODULE AND THE POLYGON MODULE WILL BE
PRNT PROMPTED IN A NEW FORMAT. WHEN THE PROGRAM PROMPTS FOR AN INPUT
PRNT VALUE, THE CURRENT VALUE WILL BE DISPLAYED TO THE RIGHT OF THE
PRNT INPUT PROMPT MESSAGE. THIS WILL APPEAR AS :
PRNT
PRNT                    'INPUT PROMPT:    = CURRENT VALUE'
PRNT
PRNT THIS WILL FACILITATE CHANGES TO INPUT DATA IF MORE THAN ONE ITER-
PRNT ATION THROUGH THE MODULE(S) IS REQUIRED.
*START CONTINUE
LT=0
VT=0
J=1
REPEAT
PRNT
PRNT GROUP [J] LCG:
READ(VALUE?)LC(J)
PRNT GROUP [J] VCG:
READ(VALUE?)VC(J)
LM(J)=W(J)*LC(J)
VM(J)=W(J)*VC(J)
LT=LT+LM(J)
VT=VT+VM(J)
J=J+1
UNTIL(J.GT.7)
LC(8)=LT/W(8)
VC(8)=VT/W(8)
LM(8)=LT
VM(8)=VT
PRNT
READ(LCG OF VARIABLE LOAD?)LC(11)
READ(VCG OF VARIABLE LOAD?)VC(11)
PRNT
LM(11)=W(11)*LC(11)
VM(11)=W(11)*VC(11)
PRNT TOTAL LEAD IS [W(9)] TONS.
PRNT
READ(AMOUNT OF LEAD FOR MARGIN IN TONS?)ML
SL=W(9)-ML
```

```
LSL=(W(12)*LCBNSC-LM(8)-LM(11)-ML*(LOA/2))/SL
LC(9)=(SL*LSL+ML*(LOA/2))/W(9)
PRNT REQUIRED STABILITY LEAD LCG IS [LSL].
READ(IS THIS FEASIBLE <Y/N>? )&R
IF(&R.EQ."N") GOTO START
LM(9)=W(9)*LC(9)
VM(9)=SL*6+ML*R
VC(9)=VM(9)/W(9)
LM(10)=LM(8)+LM(9)
VM(10)=VM(8)+VM(9)
LC(10)=LM(10)/W(10)
VC(10)=VM(10)/W(10)
LM(12)=LM(10)+LM(11)
VM(12)=VM(10)+VM(11)
VC(13)=R
VM(13)=R*W(13)
LC(13)=(W(14)*LCBSUB-LM(12))/W(13)
PRNT
PRNT REQUIRED MBT LCG IS [LC(13)].
READ(IS THIS FEASIBLE <Y/N>? )&R
IF(&R.EQ."N") GOTO START
LM(13)=W(13)*LC(13)
LM(14)=LM(12)+LM(13)
LC(14)=LM(14)/W(14)
VM(14)=VM(12)+VM(13)
VC(14)=VM(14)/W(14)
BG=R-VC(14)
PRNT
PRNT SUBMERGED STABILITY (BG) IS [BG] FT.
READ(IS THIS ACCEPTABLE <Y/N>? )&R
IF(&R.EQ."N") GOTO START
PRNT THE OUTPUT FOR THIS MODULE WILL BE IN FILE "MIT.BALOUT".
J=1
REPEAT
VY(J)=VC(J)-R
J=J+1
UNTIL(J.GT.14)
OPENW 3,"MIT.BALOUT"
&X="RESULTS FROM BALANCE MODULE"
WRITEF 3,&X
&P1="     "
&P2="   "
&P3="  "
WRITEF 3,&P1
&X="GROUP  WEIGHT     LCG     VCG"
WRITEF 3,&X
WRITEF 3,&P1
J=1
REPEAT
&J=J
&X=W(J)
&W=&X(1,8)
&X=LC(J)
&LC=&X(1,6)
&X=VC(J)
&VC=&X(1,6)
&OUT=" "+&J+"    "+&W+&P2+&LC+&P3+&VC
WRITEF 3,&OUT
J=J+1
UNTIL(J.GT.7)
```

```
J=8
&G=" A1    "
GOSUB IO
J=9
&G=" LEAD "
GOSUB IO
J=10
&G=" A     "
GOSUB IO
J=11
&G=" V LD "
GOSUB IO
J=13
&G=" MBT  "
GOSUB IO
&NSC=W(12)
&SUB=W(14)
&LCBN=LCBNSC
&LCBS=LCBSUB
&BG=BG
WRITEF 3,&P1
&SL=SL
&ML=ML
&OUT="MARGIN LEAD (TONS): "+&ML+" @VCG=D/2, LCG=LOA/2"
WRITEF 3,&OUT
&OUT="STABILITY LEAD (TONS): "+&SL+" @VCG= 6 FT"
WRITEF 3,&OUT
&LSL=LSL
&OUT="STABILITY LEAD REQUIRED LCG: "+&LSL
WRITEF 3,&OUT
WRITEF 3,&P1
&OUT="NSC (TONS): "+&NSC+" @LCB: "+&LCBN
WRITEF 3,&OUT
WRITEF 3,&P1
&OUT="SUBM (TONS): "+&SUB+" @LCB: "+&LCBS
WRITEF 3,&OUT
WRITEF 3,&P1
&OUT="SUBMERGED STABILITY (BG): "+&BG+" FT"
WRITEF 3,&OUT
OPENW 4,"MIT.PASSPOLY"
&X=LCBSUB
WRITEF 4,&X
&X=LC(11)
WRITEF 4,&X
&X=LC(13)
WRITEF 4,&X
GOTO PLOT
#IO CONTINUE
&X=W(J)
&W=&X(1,8)
&X=LC(J)
&LC=&X(1,6)
&X=VC(J)
&VC=&X(1,6)
&OUT=&G+&W+&P2+&LC+&P3+&VC
WRITEF 3,&OUT
RTNSUB
#PLOT
SELECT MODE MODEL
SELECT LAYER 3
```

```
J=1
REPEAT
&J=J
INSERT POINT:X[LC(J)]Y[VY(J)]<CR>
XP=LC(J)+1
INSERT TEXT [&J] HGT 1.5:X[XP]Y[VY(J)]<CR>
J=J+1
UNTIL(J.GT.7)
INSERT POINT:X[LC(9)]Y[VY(9)]<CR>
XP=LC(9)+1
INSERT TEXT 'LEAD' HGT 1.5:X[XP]Y[VY(9)]<CR>
INSERT POINT:X[LC(13)]Y[VY(13)]<CR>
XP=LC(13)+1
INSERT TEXT 'MBT' HGT 1.5:X[XP]Y[VY(13)]<CR>
RUN NEW MIT.POLY
END
```

MIT.&BCD.POLY
5- 7-84 10:14:02 FUTIL 6.21

```
DIM W(16),XV(10),YV(10),XP(10),YP(10),XT(10),YT(10),M(8),Y(8)
OPENR 1,"MIT.BALPASS"
J=1
REPEAT
READF 1,&X
W(J)=&X
J=J+1
UNTIL(J.GT.16)
VLD=W(11)
MBT=W(13)
DSUB=W(14)
OPENR 2,"MIT.PASSPOLY"
READF 2,&X
LSUB=&X
READF 2,&X
LVLD=&X
READF 2,&X
LMBT=&X
#START CONTINUE
BAL=VLD
PRNT TOTAL VARIABLE LOAD IS [VLD TONS.
PRNT
READ(WEIGHT OF CREW AND EFFECTS?)CE
READ(LCG?)LCE
BAL=BAL-CE
GOSUB STAT
READ(WEIGHT OF SLBM'S OR COMPENSATING WATER?)MC
READ(LCG?)LMC
BAL=BAL-MC
GOSUB STAT
READ(WEIGHT OF SANITARY TANKS AND WATER?)ST
READ(LCG?)LST
BAL=BAL-ST
GOSUB STAT
READ(WEIGHT OF LUBE OIL IN SUMPS?)LO
READ(LCG?)LLO
BAL=BAL-LO
GOSUB STAT
READ(WEIGHT OF FIXED CLFAN F. O. <IE. SHIELD TANK>?)CFO
READ(LCG?)LCFO
BAL=BAL-CFO
GOSUB STAT
READ(WEIGHT OF PROVISIONS AND STORES?)PS
READ(LCG?)LPS
BAL=BAL-PS
GOSUB STAT
READ(WEIGHT OF REVITALIZATION OXYGEN?)O2
READ(LCG?)LO2
BAL=BAL-O2
GOSUB STAT
READ(WEIGHT OF TORPEDOS IN FORWARD ROOM?)TF
READ(LCG?)LTF
BAL=BAL-TF
GOSUB STAT
READ(WEIGHT OF TORPEDOS IN AFT ROOM?)TA
READ(LCG?)LTA
```

```
57!BAL=BAL-TA
58!GOSUB STAT
59!READ(WEIGHT OF TACTICAL MISSILES FORWARD?)MF
60!READ(LCG?)LMF
61!BAL=BAL-MF
62!GOSUB STAT
63!READ(WEIGHT OF TACTICAL MISSILES AFT?)MA
64!READ(LCG?)LMA
65!BAL=BAL-MA
66!GOSUB STAT
67!TM=TF+TA+MF+MA
68!LTM=(TF*LTF+TA*LTA+MF*LMF+MA*LMA)/TM
69!READ(WEIGHT OF WATER FOR WRT TANKS?)WRT
70!READ(LCG?)LWRT
71!BAL=BAL-WRT
72!GOSUB STAT
73!READ(WEIGHT OF RESERVE ELECTROLYTE?)EL
74!READ(LCG?)LEL
75!BAL=BAL-EL
76!GOSUB STAT
77!READ(WEIGHT OF FRESH WATER <POTTABLE.FEED.BATTERY RESERVE>?)FW
78!READ(LCG?)LFW
79!BAL=BAL-FW
80!GOSUB STAT
81!READ(WEIGHT OF RESERVE LUBE & HYDRAULIC OIL?)RLO
82!READ(LCG?)LRLO
83!BAL=BAL-RLO
84!GOSUB STAT
85!PRNT THE LAST VARIABLE LOAD ITEM TO BE CONSIDERED IS FUEL OIL. TO ENSURE
86!PRNT A MARGIN OF SAFETY AT THE BASE OF THE POLYGON, A VARIABLE FUEL OIL
87!PRNT (VFO) TANK IS AUTOMATICALLY SPECIFIED. THE REQUIRED VFO CAPACITY IS
88!PRNT COMPUTED USING YOUR INPUT OF THE FRACTION OF TOTAL FUEL TO
89!PRNT PLACE INTO THE VFO TANK. THE MINIMUM SUGGESTED FRACTION IS
90!PRNT .23 (THE RATIO DIFFERENTIAL OF FUEL AND WATER DENSITIES).
91!PRNT IT IS HIGHLY RECOMMENDED THAT AN ADDITIONAL MARGIN OF AT
92!PRNT LEAST .05 BE SPECIFIED TO ALLOW A SAFETY MARGIN AT THE BASE
93!PRNT OF THE POLYGON.
94!PRNT
95!READ(FRACTION OF FUEL TO BE PLACED IN VFO?) VFR
96!TFO=BAL
97!VFO=VFR*TFO
98!FBT=TFO-VFO
99!FL=CE+MC+ST/3+LO+CFO
100!LFL=(CE*LCE+MC*LMC+ST*LST+LO*LLO+CFO*LCFO)/FL
101!PRNT FBT CAPACITY (TONS): [FBT]
102!PRNT VFO CAPACITY (TONS): [VFO]
103!PRNT
104!READ(AMOUNT OF FBT <TONS> IN FORWARD TANKS?)FBTF
105!FBTA=FBT-FBTF
106!PRNT
107!READ(FWD FBT LCG?)LFBTF
108!READ(AFT FBT LCG?)LFBTA
109!LFBT=(FBTF*LFBTF+FBTA*LFBTA)/FBT
110!READ(VFO LCG?)LVFO
111!PRNT
112!PRNT
113!PRNT
114!PRNT
115!PRNT VARIABLE LOAD RECAP:
116!PRNT    ITEM              WEIGHT         LCG
```

```
117!PRNT
118!PRNT CREW & EFFECTS       [CE]           [LCE]
119!PRNT SLBM'S              [MC]           [LMC]
120!PRNT SANITARY TANKS      [ST]           [LST]
121!PRNT SUMP LUBE OIL       [LO]           [LLO]
122!PRNT CLEAN F.O. (FIXED)  [CFO]          [LCFO]
123!PRNT ***TOTAL FIXED***   [FL]           [LFL]
124!PRNT PROV. & STORES      [PS]           [LPS]
125!PRNT REVITALIZATION O2   [O2]           [LO2]
126!PRNT TORPEDOS FWD ROOM   [TF]           [LTF]
127!PRNT TORPEDOS AFT ROOM   [TA]           [LTA]
128!PRNT TACT. MISSILES FWD  [MF]           [LMF]
129!PRNT TACT. MISSILES AFT  [MA]           [LMA]
130!PRNT WRT TANKS           [WRT]          [LWRT]
131!PRNT RESERVE ELECTROLYTE [EL]           [LEL]
132!PRNT TOTAL FRESH WATER   [FW]           [LFW]
133!PRNT RESERVE LUBE OIL    [RLO]          [LRLO]
134!PRNT FUEL BALLAST FWD    [FBTF]         [LFBTF]
135!PRNT FUEL BALLAST AFT    [FBTA]         [LFBTA]
136!PRNT VARIABLE FUE     .   [VFO]          [LVFO]
137!READ(ARE THESE VALUES ACCEPTABLE <Y/N>? )&R
138!IF(&R.EQ."NO") GOTO START
139!PRNT
140!AFL=LSUB-LFL
141!APS=LSUB-LPS
142!AO2=LSUB-LO2
143!ATM=LSUB-LTM
144!AWRT=LSUB-LWRT
145!AEL=LSUB-LEL
146!AFW=LSUB-LFW
147!ARLO=LSUB-LRLO
148!AFBT=LSUB-LFBT
149!AVFO=LSUB-LVFO
150!ATF=LSUB-LTF
151!ATA=LSUB-LTA
152!AMF=LSUB-LMF
153!AMA=LSUB-LMA
154!AFBTF=LSUB-LFBTF
155!AFBTA=LSUB-LFBTA
156!MFL=FL*AFL
157!MPS=PS*APS
158!MO2=O2*AO2
159!MTM=TM*ATM
160!MWRT=WRT*AWRT
161!MEL=EL*AEL
162!MFW=FW*AFW
163!MRLO=RLO*ARLO
164!MFBT=FBT*AFBT
165!MVFO=VFO*AVFO
166!MTF=TF*ATF
167!MTA=TA*ATA
168!MMF=MF*AMF
169!MMA=MA*AMA
170!MFBTF=FBTF*AFBTF
171!MFBTA=FBTA*AFBTA
172!CVL=.99375
173!CVH=1.009375
174!GOTO NSTM
175!*STAT CONTINUE
176!PRNT
```

```
177!PRNT BALANCE OF VARIABLE LOAD ACCOUNT: (BAL)
178!PRNT
179!RTNSUB
180!#NSTM CONTINUE
181!LDSUB=W(14)-W(10)
182!#HEAV1 CONTINUE
183!CW=43/35
184!H1=FL+TM+.5*PS+EL+WRT+.5*RLO+.5*FW+CW*FBT
185!MH1=MFL+MTM+.5*MPS+MEL+MWRT+.5*MRLO+.5*MFW+CW*MFBT
186!VBH1=LDSUB-MBT-H1
187!AMBT=LSUB-LMBT
188!MMBT=MBT*AMBT
189!MVBH1=-MMBT-MH1
190!XV(1)=MVBH1*CVL
191!YV(1)=VBH1*CVL
192!#HEAV2 CONTINUE
193!H2=FL+TM+PS+EL+WRT+RLO+FW+CW*FBT
194!MH2=MFL+MTM+MPS+MEL+MWRT+MRLO+MFW+CW*MFBT
195!VBH2=LDSUB-MBT-H2
196!MVBH2=-MMBT-MH2
197!XV(2)=MVBH2*CVL
198!YV(2)=VBH2*CVL
199!#LITE1
200!L1=FL+.75*PS+WRT+.75*RLO+.75*FW+FBT+VFO
201!ML1=MFL+.75*MPS+MWRT+.75*MRLO+.75*MFW+MFBT+MVFO
202!VBL1=LDSUB-MBT-L1
203!MVBL1=-MMBT-ML1
204!XV(3)=MVBL1*CVH
205!YV(3)=VBL1*CVH
206!#LITE2
207!L2=FL+WRT+RLO+VFO+CW*FBT
208!ML2=MFL+MWRT+MRLO+MVFO+CW*MFBT
209!VBL2=LDSUB-MBT-L2
210!MVBL2=-MMBT-ML2
211!XV(4)=MVBL2*CVH
212!YV(4)=VBL2*CVH
213!#HF1
214!HF1=FL+MF+TF+.75*PS+EL+WRT+.75*RLO+.75*FW+VFO+FBTA+CW*FBTF
215!MHF1=MFL+MTF+MMF+.75*MPS+MEL+MWRT+.75*MRLO+.75*MFW+MVFO+MFBTA+CW*MFBTF
216!VBHF1=LDSUB-MBT-HF1
217!MVBHF1=-MMBT-MHF1
218!XV(5)=MVBHF1
219!YV(5)=VBHF1
220!#HF2
221!HF2=FL+MF+TF+.5*PS+EL+WRT+.25*RLO+.5*FW+VFO+FBTA+CW*FBTF
222!MHF2=MFL+MTF+MMF+.75*MPS+MEL+MWRT+.25*MRLO+.5*MFW+MVFO+MFBTA+CW*MFBTF
223!VBHF2=LDSUB-MBT-HF2
224!MVBHF2=-MMBT-MHF2
225!XV(6)=MVBHF2
226!YV(6)=VBHF2
227!#HA
228!HA=FL+MA+TA+.5*PS+WRT+.75*RLO+.5*FW+VFO+FBTF+CW*FBTA
229!MHA=MFL+MMA+MTA+.5*MPS+MWRT+.75*MRLO+.5*MFW+MVFO+MFBTF+CW*MFBTA
230!VBHA=LDSUB-MBT-HA
231!MVBHA=-MMBT-MHA
232!XV(7)=MVBHA
233!YV(7)=VBHA
234!#N
235!N=H2
236!MN=MH2
```

```
VBN=LDSUB-MBT-N
MVBN=-MMBT-MN
XV(8)=MVBN
YV(8)=VBN
#CONDM CONTINUE
CM=H2-CW*FBT+FBT
MCM=MH2-CW*MFBT+MFBT
VBCM=LDSUB-MBT-CM
MVBCM=-MMBT-MCM
XV(9)=MVBCM
YV(9)=VBCM
<#************** SET UP LOCATION OF TRIM & AUX TANKS ***************
PRNT
PRNT YOUR ACTUAL DESIGN MAY HAVE AS MANY TRIM AND AUX TANKS AS YOU DEEM
PRNT APPROPRIATE FOR YOUR PARTICULAR ARRANGEMENT SCHEME. TO SIMPLIFY THE
PRNT CALCULATION PROCESS FOR THE EQUILIBRIUM POLYGON, YOU ARE ASKED TO
PRNT ENTER AGGREGATE CAPACITY AND LCG FOR THE FORWARD TRIM GROUP. THE
PRNT AFTER TRIM GROUP, AND THE AUXILIARY GROUP TANKS.
PRNT
GOTO SIZE
#LOOP
EXECV
READ(LCG OF FORWARD TRIM TANK GROUP?)LFT
READ(LCG OF AFTER TRIM TANK GROUP?)LAT
READ(LCG OF AUX TANK GROUP?)LAUX
AFT=LSUB-LFT
AAT=LSUB-LAT
AAUX=LSUB-LAUX
RTNSUB
#SIZE CONTINUE
J=1
REPEAT
IF(XV(J).GT.5000)GOTO LARGE
IF(XV(J).LT.-5000)GOTO LARGE
IF(YV(J).GT.500)GOTO LARGE
J=J+1
UNTIL(J.GT.9)
GOTO SMALL
#LARGE
DEL PAR MIT.HULLOUT
EXIT PART FILE MIT.HULLOUT
XS=1000
YS=100
ACT PAR MIT.POLYOUT
ACT DRA POLYGON FORM MIT.LARGEPOLY DRAW POLY
SEL CPL LEFT
DEF VIE GRAPH: X11Y6.X0Y0.X22Y17 <CR>
GOTO GRAPH
#SMALL CONTINUE
DEL PAR MIT.HULLOUT
EXIT PART FILE MIT.HULLOUT
XS=500
YS=50
DEL PAR MIT.POLYOUT
ACT PAR MIT.POLYOUT
ACT DRA POLYGON FORM MIT.SMALLPOLY DRAW POLY2
SEL CPL LEFT
DEF VIE GRAPH: X11Y6.X0Y0.X22Y17 <CR>
#GRAPH CONTINUE
J=1
```

```
297!REPEAT
298!XP(J)=XV(J)/XS
299!YP(J)=YV(J)/YS
300!INSERT POINT: X[XP(J)]Y[YP(J)] <CR>
301!XT(J)=XP(J)+.2
302!YT(J)=YP(J)-.1
303!J=J+1
304!UNTIL(J.GT.9)
305!INS TEX H1 HG .2: X[XT(1)]Y[YT(1)] <CR>
306!INS TEX H2 HG .2: X[XT(2)]Y[YT(2)] <CR>
307!INS TEX L1 HG .2: X[XT(3)]Y[YT(3)] <CR>
308!INS TEX L2 HG .2: X[XT(4)]Y[YT(4)] <CR>
309!INS TEX HF1 HG .2: X[XT(5)]Y[YT(5)] <CR>
310!INS TEX HF2 HG .2: X[XT(6)]Y[YT(6)] <CR>
311!INS TEX HA HG .2: X[XT(7)]Y[YT(7)] <CR>
312!INS TEX N HG .2: X[XT(8)-.5]Y[YT(8)] <CR>
313!INS TEX M HG .2: X[XT(9)]Y[YT(9)] <CR>
314!GOSUB LOOP
315!GOTO PLTNK
316!#CAP CONTINUE
317!EXECV
318!READ(CAPACITY <TONS> OF FORWARD TRIM GROUP?)CFT
319!READ(CAPACITY <TONS> OF AUXILIARY GROUP?)CAUX
320!READ(CAPACITY <TONS> OF AFTER TRIM GROUP?)CAT
321!RTNSUB
322!#PLTNK CONTINUE
323!GOSUB CAP
324!M(1)=0
325!M(2)=AFT*CFT
326!M(3)=M(2)+AAUX*CAUX
327!M(4)=M(3)+AAT*CAT
328!M(5)=M(4)-AFT*CFT
329!M(6)=M(5)-AAUX*CAUX
330!M(7)=0
331!Y(1)=0
332!Y(2)=CFT
333!Y(3)=CAUX+CFT
334!Y(4)=Y(3)+CAT
335!Y(5)=Y(4)-CFT
336!Y(6)=Y(5)-CAUX
337!Y(7)=0
338!J=1
339!REPEAT
340!M(J)=M(J)/XS
341!Y(J)=Y(J)/YS
342!J=J+1
343!UNTIL(J.GT.7)
344!INS LIN TAG=1: X[M(1)]Y[Y(1)],X[M(2)]Y[Y(2)], <CR>
345!INS LIN TAG=2: X[M(2)]Y[Y(2)],X[M(3)]Y[Y(3)], <CR>
346!INS LIN TAG=3: X[M(3)]Y[Y(3)],X[M(4)]Y[Y(4)], <CR>
347!INS LIN TAG=4: X[M(4)]Y[Y(4)],X[M(5)]Y[Y(5)], <CR>
348!INS LIN TAG=5: X[M(5)]Y[Y(5)],X[M(6)]Y[Y(6)], <CR>
349!INS LIN TAG=6: X[M(6)]Y[Y(6)],X[M(7)]Y[Y(7)], <CR>
350!EXECV
351!READ(IS THIS POLYGON SATISFACTORY <Y/N> ? )AR
352!IF(&R.EQ."Y") GOTO FINAL
353!DEL ENT: TAG 1,TAG 2,TAG 3,TAG 4,TAG 5,TAG 6 <CR>
354!EXECV
355!READ(CHANGE CAPACITIES ONLY <1>, OR LCG'S & CAPACITIES <2> ? )CHC
356!IF(CHC.EQ.2) GOTO BOTH
```

```
IF(CHC.EQ.1) GOTO PLTNK
#BOTH CONTINUE
GOSUB LOOP
GOTO PLTNK
#FINAL CONTINUE
OPENW 3,"MIT.PASS.POLYDAT"
&X=CE
GOSUB 0
&X=LCE
GOSUB 0
&X=MC
GOSUB 0
&X=LMC
GOSUB 0
&X=ST
GOSUB 0
&X=LST
GOSUB 0
&X=LO
GOSUB 0
&X=LLO
GOSUB 0
&X=CFO
GOSUB 0
&X=LCFO
GOSUB 0
&X=FL
GOSUB 0
&X=LFL
GOSUB 0
&X=PS
GOSUB 0
&X=LPS
GOSUB 0
&X=O2
GOSUB 0
&X=LO2
GOSUB 0
&X=TF
GOSUB 0
&X=LTF
GOSUB 0
&X=TA
GOSUB 0
&X=LTA
GOSUB 0
&X=MF
GOSUB 0
&X=LMF
GOSUB 0
&X=MA
GOSUB 0
&X=LMA
GOSUB 0
&X=WRT
GOSUB 0
&X=LWRT
GOSUB 0
&X=EL
GOSUB 0
```

```
&X=LEL
GOSUB O
&X=FW
GOSUB O
&X=LFW
GOSUB O
&X=RLO
GOSUB O
&X=LRLO
GOSUB O
&X=FBTF
GOSUB O
&X=LFBTF
GOSUB O
&X=FBTA
GOSUB O
&X=LFBTA
GOSUB O
&X=VFO
GOSUB O
&X=LVFO
GOSUB O
&X=CFT
GOSUB O
&X=LFT
GOSUB O
&X=CAUX
GOSUB O
&X=LAUX
GOSUB O
&X=CAT
GOSUB O
&X=LAT
GOSUB O
!PLOT DOT SCALE .5
EXIT PART F
GOTO END
#O
WRITEF 3.&X
RTNSUB
#END
RUN NEW MIT.POLYFILE
END
```

MIT.&BCD.POLYFILE
5- 7-84 19:22:03 FUTIL 6.21

```
OPENR 1,"MIT.PASS.POLYDAT"
OPENW 2,"MIT.OUT.POLY"
PRNT OUTPUT FOR THE POLYGON MODULE WILL BE IN 'MIT.OUT.POLY'.
&T="POLYGON SUMMARY OUTPUT FILE"
WRITEF 2,&T
&P="       "
WRITEF 2,&P
WRITEF 2,&P
&T="     ITEM                    LCG                WEIGHT"
WRITEF 2,&T
WRITEF 2,&P
READF 1,&X
READF 1,&Y
&T="CREW AND EFFECTS         "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="SLBM'S OR COMP. WATER    "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="SANITARY TANKS           "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="LUBE OIL IN SUMPS        "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="FIXED CLEAN FUEL OIL     "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="****TOTAL FIXED ITEMS**** "+&Y+"             "+&X
WRITEF 2,&T
GOSUB RD
&T="PROVISIONS AND STORES    "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="REVITALIZATION O2        "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="TORPEDOS IN FWD ROOM     "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="TORPEDOS IN AFT ROOM     "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="TACTICAL MISSILES FWD    "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="TACTICAL MISSILES AFT    "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="WRT TANKS                "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="RESERVE ELECTROLYTE      "+&Y+"              "+&X
WRITEF 2,&T
GOSUB RD
&T="TOTAL FRESH WATER        "+&Y+"              "+&X
```

```
57!WRITEF 2,&T
58!GOSUB RD
59!&T="RESERVE LUBE OIL          "+&Y+"           "+&X
60!WRITEF 2,&T
61!GOSUB RD
62!&T="FUEL BALLAST FWD          "+&Y+"           "+&X
63!WRITEF 2,&T
64!GOSUB RD
65!&T="FUEL BALLAST AFT          "+&Y+"           "+&X
66!WRITEF 2,&T
67!GOSUB RD
68!&T="VARIABLE FUEL OIL         "+&Y+"           "+&X
69!WRITEF 2,&T
70!GOSUB RD
71!WRITEF 2,&P
72!WRITEF 2,&P
73!&T="VARIABLE BALLAST TANK DATA:"
74!WRITEF 2,&T
75!&T="   TANK GROUP             LCG           CAPACITY (TONS)"
76!WRITEF 2,&T
77!&T="FORWARD TRIM                "+&Y+"            "+&X
78!WRITEF 2,&T
79!GOSUB RD
80!&T="AUXILIARY                   "+&Y+"            "+&X
81!WRITEF 2,&T
82!GOSUB RD
83!&T="AFTER TRIM                  "+&Y+"            "+&X
84!WRITEF 2,&T
85!GOTO END
86!#RD CONTINUE
87!READF 1,&X
88!READF 1,&Y
89!RTNSUB
90!#END
91!END
```